# Video Based Motion Trajectory Analysis of Pedestrians

Arun Kumar Chandran

National University of Singapore

2015

# Video Based Motion Trajectory Analysis of Pedestrians

**Arun Kumar Chandran**
*B.Eng., Anna University, India*

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

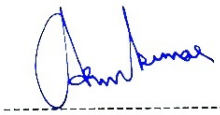
Department of Electrical and Computer Engineering
National University of Singapore

2015

# Declaration

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in this thesis.

This thesis has also not been submitted for any degree in any university previously.



Arun Kumar Chandran

Date: 19, March, 2015

# Acknowledgments

I thank my supervisors, Assoc Prof. Loh Ai Poh and Assoc Prof. Prahlad Vadakkepat for their support, guidance and most importantly their patience during my years at NUS. They taught me how to do quality research in a systematic and thorough manner. They will always be a source of inspiration for me, both in professional and personal life.

I thank the National University of Singapore (NUS) for the financial support to carry out this research. A very special thanks goes to Prof.Lawrence Wai-Choong Wong for offering me an opportunity to work with his research team to develop interesting research applications. I should again thank my supervisors for introducing me to Prof. Wong. I thank the team at NUS Ambient Intelligence Lab for their support and fruitful discussions. Especially, thanks to Junius (Soh Hock Heng), Lue Ik Hong. My seniors'(Dr. Pramod Kumar, Dr. Andras Gabor Kupscik) guidance and support deserves a special mention here. Special thanks to Yap Wei Rong, Yee Yee and Loh Hui Ying for their help in translating one of the algorithms into a C++ implementation. I wish to say a big thank you to my friend Hari, who planted the idea of a PhD in NUS.

I owe my family a huge thanks for their unconditional support throughout my life and for always being on my side. This acknowledgement will no be complete without thanking my parents. It was a tough journey but it would have been harder without their brilliant guidance and emotional support. It was difficult without them but I always felt their continuous interaction, support and love put me across all the tough days I have faced. I thank my sister and niece who cheered me up with their smile in my tough times.

# Contents

# *Abstract*

Video cameras are widely used for surveillance in public domains, which allow security personnel to remotely monitor the activities in an area of interest. Due to inadequate manpower, effective manual monitoring of multiple video feeds is not possible. With progress in video processing and computing technologies, it is possible to develop real time algorithms to intelligently detect and decipher scenes in the video frames. In public domains like bus and rail interchanges, shopping malls, numerous applications can be derived from identifying pedestrian crowds. Pedestrian crowds can be analyzed by tracking the movement of individuals to understand the crowd behavior. Automatic group identification and tracking can highlight regions of interest from numerous video feeds, thereby aiding the security personnel in the surveillance of the area.

In this thesis, a three-level blob filter utilizing the pedestrian detections from Histogram of Oriented Gradients for human detection and Background Subtraction is developed to detect pedestrians. The three-level blob filter addresses pedestrian detection challenges such as illumination variations and pedestrian-like confusers.

A method to identify social groups of pedestrians from pedestrian crowd videos is proposed. The method identifies social groups based on the social psychological theory of pedestrian groups, by monitoring whether the pedestrians move closely together in the same direction with almost the same velocity for a certain period of time. The pedestrian group features such as group size, group member locations, time of group identification are extracted and stored in a proposed data structure, termed as Pedestrian Group Record.

Several applications which use the information in the Pedestrian Group Record are proposed. They are Real-time pedestrian meeting-event detection system, Stall pedestrian occupancy determination, People Counting at queuing regions and door locations, Crowd density estimation at train platforms and Visual representations of pedestrian's group history. The proposed methods and the applications are validated using real-world video data sets. Some of the applications are deployed in real-world locations after the validation.

Pedestrian motion changes which happen during the convergence of pedestrians to form a group is utilized to predict future pedestrian groups. In this proposed method, the pair-

wise pedestrian motion changes which lead to group formations are learnt by a Support Vector Machine. The learnt model is utilized to predict new pairwise pedestrian motion change. A similar approach is proposed to predict approach of potential customers to food outlets.

# List of Tables

# List of Figures

# List of Acronyms

| | |
|---|---|
| HOG | Histogram of Oriented Gradients |
| BS | Background Subtraction |
| NMSC | Non-recursive Motion Similarity Clustering |
| PGR | Pedestrian Group Record |
| NUS | National University of Singapore |
| SVM | Support Vector Machines |
| NMSC | Non-recursive Motion Similarity based Clustering |
| NUS | National University of Singapore |
| PDF | Probabilistic Density Function |
| MHT | Multi Hypothesis Tracking |
| SFM | Social Force Model |
| GPU | Graphics Processing Unit |
| LCSS | Longest Common Sub Sequence |
| TGP | Temporal Grouping Plot |
| CCTV | Closed Circuit Television |
| IP | Internet Protocol |
| RPMVIS | Real-time Pedestrian Meetings and Visits Identification System |
| SDK | Software Development Kit |
| FPS | Frames Per Second |
| AMI | Ambient Intelligence Lab |
| IDMI | Interactive Digital Media Institute |
| ROI | Region Of Interest |
| MRT | Mass Rapid Transit |
| RBF | Radial basis function |

# Notation and Symbols

| | |
|---|---|
| $t$ | time |
| $z_t$ | 2 dimensional coordinate |
| $\hat{z}_{t\|t-1}$ | predicted coordinate |
| $F_i^t$ | pedestrian trajectory |
| $(u_t, v_t)$ | velocity components in $x$ and $y$ dimensions |
| $D_B$ | blobs detected from background subtraction |
| $D_H$ | blobs detected from Histogram of Oriented Gradients |
| $\sigma$ | standard deviation |
| $\mu$ | mean |
| $s_\epsilon$ | distance between pedestrians |
| $v_\epsilon$ | relative speed difference |
| $d_\epsilon$ | relative difference in direction of travel |
| $V$ | velocity |
| $T$ | trajectory length |
| $S$ | similarity graph |
| $R$ | correlation matrix |
| $Cl$ | pedestrian cluster |
| $\kappa$ | kappa score |
| $E_s$ | ensemble matrix |
| $G$ | spatial grid |
| $\bar{x}$ | feature vector |
| $\bar{X}$ | feature vector set |

**I dedicate this thesis to my beloved family.**

# Chapter 1

# Introduction

Video surveillance in public domains has gained attention at a time when security is a major concern. The generated video feeds allow security personnel to remotely monitor activities in an area of interest. One of the problems with such monitoring systems is that often times, there is inadequate manpower to effectively monitor multiple video feeds manually. With progress in video processing and computing technologies, it is possible to develop real time algorithms to intelligently detect and decipher scenes in the video frames. Such algorithms will depend on the type of scene analysis required and the locations considered. This chapter will explain the motivation for the thesis work, the overview of the social psychology ideas adopted and the major contributions of the thesis.

Public locations like schools, transportation hubs, sports venues, and public gatherings are typically characterized by a large number of people exhibiting frequent and complex social interactions [7, 8]. In order to identify activities and behaviors in such environment, it is necessary to understand the interactions taking place at a group level [9, 10, 11]. Understanding group-level interaction is particularly important in surveillance and security, where gang related activities are the root cause of most criminal behaviors and disorderly conduct. Many works in the literature [12, 13, 14] have identified that congestion and violence are mostly created by individuals and aggravated by groups of people rather than individuals. These examples show the importance of identifying and understanding the group-level behavior, in addition to the individual behavior. Pedestrian crowds can be analyzed by tracking the movement of groups to understand the crowd behavior. Auto-

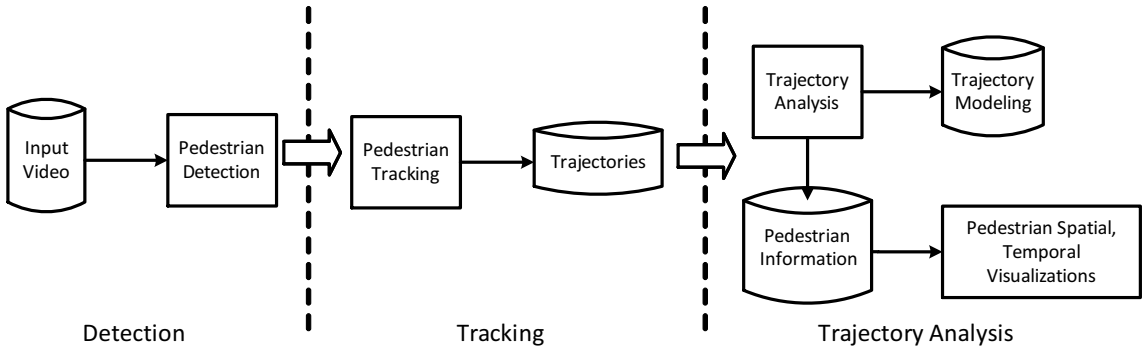



Figure 1.1: Motion Trajectory Analysis Framework with the three levels of processing (representing the complexity of the information).

matic group identification and tracking can highlight regions of interest from numerous video feeds, thereby aiding the security personnel in the surveillance of the area. Numerous applications can be derived from identifying pedestrian groups and motion patterns. Pedestrian group movements can be tracked, crowd levels at public locations such as queues, buses, train platforms can be estimated and crowd flow based on different group sizes can be identified. Moving a step further, prediction of pedestrian activities such as group formations, can be highly useful for enforcement agencies. These applications form the motivation for this thesis work, to develop novel methods and algorithms to extract group-level and pedestrian-level information from the surveillance videos and to extract higher level of information about the pedestrian crowd.

Figure 1.1 shows a framework to extract useful information about pedestrians by analyzing the pedestrians' motion trajectory. In this framework, pedestrians are detected from the input video. The detected pedestrians are tracked to build their trajectories. Trajectories are analyzed to extract useful information about the pedestrians which can be visualized. The trajectories can be modeled to generate a spatial and temporal model of the pedestrian motion. Pedestrian detection, tracking and trajectory analysis are the three major processes in this framework. In this thesis, the trajectory information is utilized to decipher the grouping behavior of these pedestrians. Locations with low ($< 3$ pedestrians per $m^2$) to medium (3 - 4 pedestrians per $m^2$) crowd density [15] are considered in this thesis.

## 1.1 Overview

According to E.T. Hall's work on proxemics [16], people tend to have different virtual spaces around them. According to this work, there are four spaces (Figure 1.2) around every pedestrian: intimate space, personal space, social space and public space. A pedestrian is considered to be related to another if he or she is at least within the social space of another pedestrian. Based on social psychological research on pedestrian groups [17], pedestrians are considered to be in a group if they are spatially within 2 meter distance of each other walking in almost the same direction with almost the same velocity. The distance value for a pedestrian group falls in the social space range (social space less than 3.6 meters in Figure 1.2) proposed by E.T. Hall. Inspired by these social psychological findings, several algorithms are proposed in this thesis to identify, predict and track pedestrian groups. Based on empirical study, these works define the conditions for existence of pedestrian groups. They do not provide information on possible (i.e. potential) pedestrian groups in future. Moving one step ahead, this thesis proposes an hypothesis to predict such future groups. The hypothesis considers the pedestrian motion changes that happen before a pedestrian group formation as unique. By learning such motion changes, future pedestrian groups could be predicted. As shown in Figure 1.3, this thesis aims to identify existing pedestrian groups as well as predict future pedestrian groups. The useful information which could be extracted out of these solutions are also explored and presented in this thesis.

## 1.2 Contributions

The major contributions of the thesis are outlined in this section.

**Automatic pedestrian group identification** Inspired by social psychological principles on group behavior [17], this method utilizes a novel Non-recursive Motion Similarity Clustering (NMSC) algorithm to cluster pedestrians based on their motion similarities. Individual pedestrians are detected and tracked in video scenes. Pedestrians are automatically clustered based on their pair-wise motion similarity by considering their relative

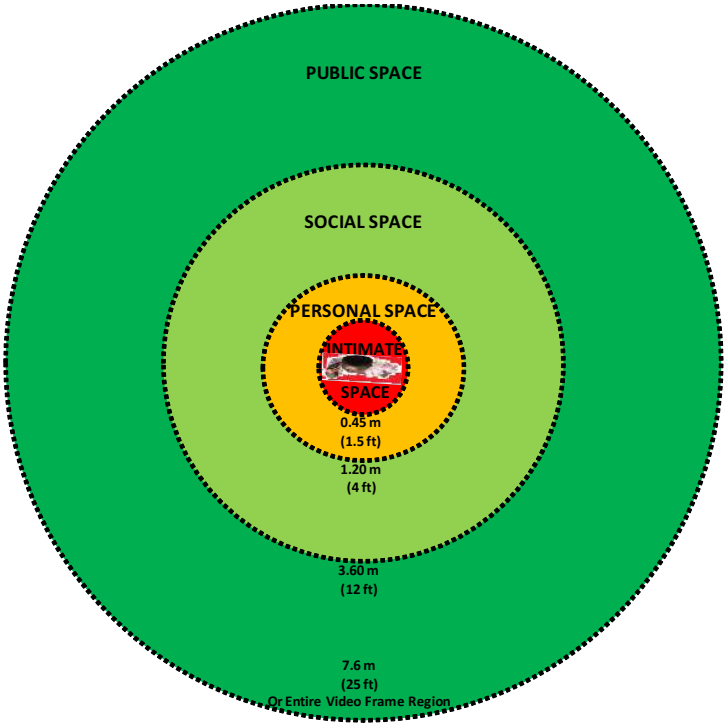



Figure 1.2: E.T. Hall's Proxemics zones.

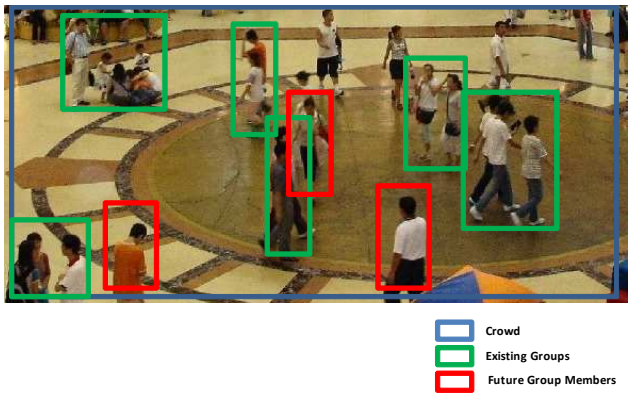



Figure 1.3: A crowd with existing pedestrian groups and future pedestrian groups.

distance, relative speed and relative direction of motion. Pedestrian clusters which persist for a period of time are identified as pedestrian groups. Performance evaluation against existing related works reveal that the proposed method identifies pedestrian groups with better accuracy and could identify large groups of more than three pedestrians (which existing related works cannot identify).

**Real-time pedestrian visits and meetings identification system** This system automatically identifies pedestrian meetings and visits from surveillance videos. The real-time system has a Pedestrian Detection and Tracking module, Pedestrian Group Identification module (NMSC algorithm), a Pedestrian Group Record (PGR) and three visualizations of the PGR information. The system uses the pedestrian group features extracted and stored in the Pedestrian Group Record (PGR) to identify the pedestrian meeting and split events. The features include group size, group member identity, group members' trajectory, detection count, absence count and time stamp at which the group is identified. A meeting event is identified when a newly formed group's existence is confirmed by checking in the pedestrian group record. A split event is identified when a group's termination is confirmed by checking again in the pedestrian group record. The real-time system has been deployed in residential halls in National University of Singapore (NUS) after performing calibrations (to address the camera-server communication lag, environmental challenges faced in pedestrian detection). The system is found to be having an event identification accuracy of greater than 80 percent.

**Applications based on pedestrian group identification** Several applications of pedestrian group identification and pedestrian detection are explored and developed. They are people counting: at queue regions, at door entrances and crowd estimation at different public modes of transport. Certain novel, learning methods are proposed to automatically detect queuing regions and to estimate crowd levels in train platforms. All these applications are based on the information which is extracted and stored in the Pedestrian Group Record (PGR). Some of the applications like people counting are deployed in real-world scenarios as pilot projects.

**Predicting future pedestrian groups and approach of potential customers** The motion changes which happen before a pedestrian group is formed are observed to be unique. These changes in pedestrian motion parameters (like distance, speed, direction of travel) during the period of transition from a being an individual to becoming a group member, are learnt by Support Vector Machines (SVM) in this method. The learnt SVM model is utilized to classify new pedestrian motion parameter variations to predict future (potential) pedestrian groups. A similar approach is adopted to predict approach of potential customers to food outlets. These methods are developed and tested with real-world cases like public locations (library), food outlets. These methods can predict the pedestrian's status ahead of time by under sampling feature vectors.

## 1.3 Thesis Outline

The thesis structure is shown in Figure 1.4. Highlights of each chapter are explained below.

**Chapter 2: Literature Survey** This chapter provides a literature survey on the tools and techniques in the background of this thesis work. These are the techniques for pedestrian detection and tracking. The state-of-the art applications of pedestrian motion trajectory analysis are also outlined in this chapter.

**Chapter 3: Preliminaries** This chapter explains the steps to extract the two-dimensional coordinates of the pedestrians from the video clips. These are image processing steps. A three-level blob filtering approach to detect pedestrians is explained with performance evaluation against existing techniques. Some of the important video data sets used in this thesis for evaluation purposes are also explained. The pin hole model to perform the translation of the three-dimensional coordinates to two-dimensional coordinates is explained.

**Chapter 4: Pedestrian Group Identification** A method to automatically identify pedestrian groups is explained in this chapter. Some of the existing related works are

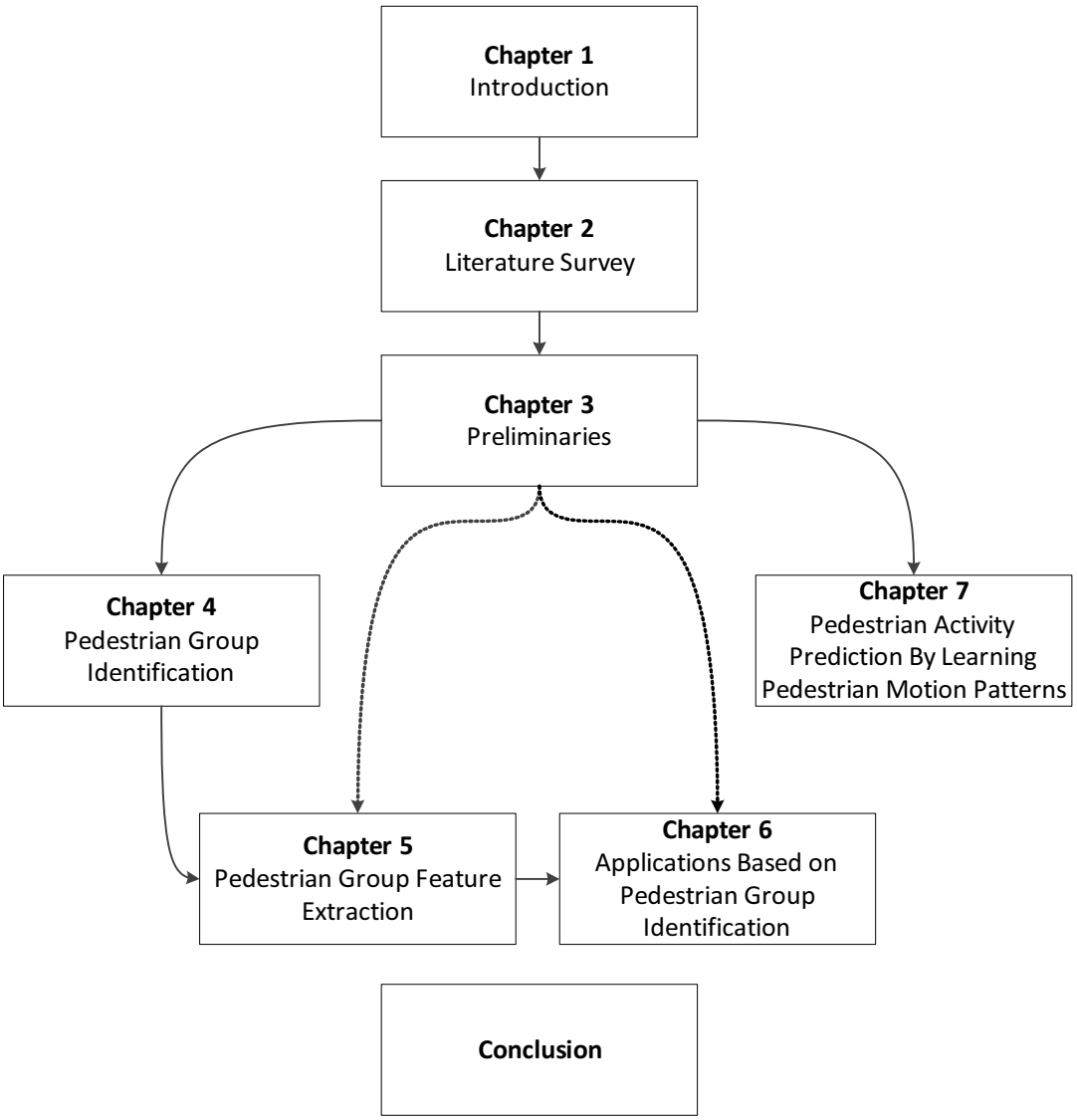



Figure 1.4: The thesis structure. The solid lines express dependency between chapters, while the dashed lines refer to related, but not directly dependent chapters.

explained with their advantages and disadvantages. The performance evaluation of the proposed method against existing related works is highlighted.

**Chapter 5: Pedestrian Group Feature Extraction** The pedestrian group features which are extracted from the process of pedestrian group identification are explained in this chapter. The data structure (pedestrian group record) utilized to store these features are also explained. A real-time system to identify pedestrian meeting events and visits is discussed. The system utilizes the group record information to automatically identify the meeting events and has several visualizations to better understand the pedestrian groups, pedestrian's group membership history and the popular areas in a monitored region. The performance evaluation of the real-time system is also discussed.

**Chapter 6: Applications Based on Pedestrian Group Identification** Several applications based on pedestrian group identification and pedestrian detection are discussed in this chapter. They are people counting at locations of queue regions, and, door entrances, crowd estimation at different public modes of transport, potential jam prediction at train networks. Certain simple, novel, learning methods are proposed: to automatically detect queuing regions and to estimate crowd levels in train platforms. Virtual environments created for some of the applications are also explained in this chapter to demonstrate the proof of concept.

**Chapter 7: Pedestrian Activity Prediction By Learning Pedestrian Motion Patterns** A process to learn pedestrian motion patterns to predict pedestrian activities such as potential customer-approach to food outlets is explained in this chapter. The motion patterns are modeled by learning the corresponding motion parameter variations using machine learning techniques such as Support Vector Machines (SVM). Two methods which adopt this process of learning are proposed in this chapter. They are prediction of potential customer-approach to food outlets and prediction of possible future pedestrian groups. Performance evaluation by comparing prediction accuracy against the ground truth is discussed. The data sets used to evaluate these methods are explained in this chapter.

**Conclusion** This chapter concludes the thesis with a summary of the work done and possible future research directions.

# Chapter 2

# Literature Survey

In recent decades, cameras are widely used for surveillance purposes. Security personnel cannot manually monitor the numerous camera video feeds to identify important security events such as illegal entry, violence, quarrels and so on. Hence, there is a growing need to develop real time algorithms to intelligently detect and decipher scenes in the video frames. Such algorithms depend on the type of locations considered and scene analysis required.

Large numbers of people visit public locations leading to complex interactions among them. In order to identify important security events in such an environment, it is necessary to understand the interactions taking place among the pedestrians at the group level. Understanding group-level interaction is particularly important in surveillance and security, because groups of people interacting with one another are generally viewed as potential threats to peace compared to a few unorganized individuals. This shows the importance of identifying and understanding the group-level behavior.

This thesis proposes novel methods and algorithms to extract pedestrian-level and group-level information from the surveillance videos. A simple framework to extract useful information about pedestrians by analyzing the pedestrian motion trajectory is explained in Figure 2.1. In this framework, pedestrians are detected from the input video. The detected pedestrians are tracked to build their trajectories. Trajectories are analyzed to extract useful information about the pedestrians which can be visualized. The trajectories can be modeled to generate a spatial and temporal model of the pedestrian motion.

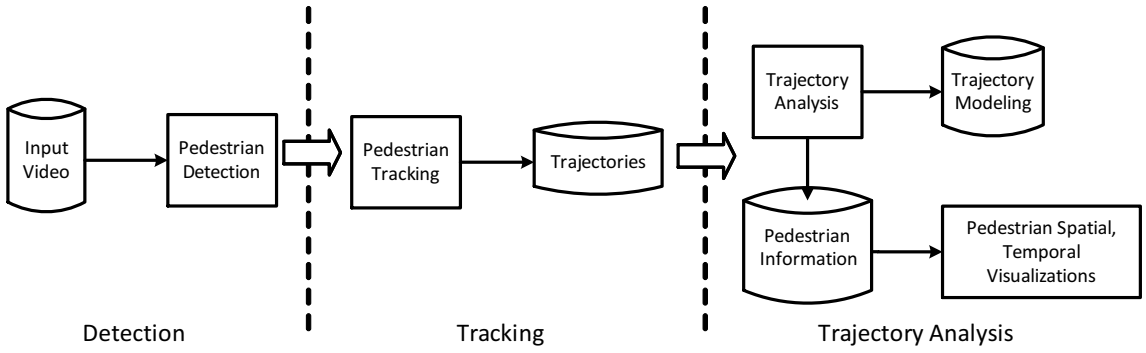



Figure 2.1: Pedestrian Motion Trajectory Analysis Framework with the three levels of processing (representing the complexity of the information).

Pedestrian detection, tracking and trajectory analysis are the three major processes in this framework.

This chapter provides a literature survey and background work on the tools and techniques used in this thesis. A review of the state-of-the-art applications which extract useful pedestrian information from their motion data is also provided. According to Figure 2.1, the survey is divided into three sections: computer vision methods for pedestrian detection and tracking, applications of pedestrian detection and tracking, and applications of pedestrian motion trajectory analysis.

The rest of the chapter is organized as follows. Section 2.1 provides a review of various computer vision methods to detect and track pedestrians. Section 2.2 highlights the standard applications of pedestrian detection and tracking. A brief survey on the state-of-the-art applications of pedestrian trajectory analysis is provided in Sec. 2.3.

## 2.1 Computer Vision Methods for Pedestrian Detection and Tracking

In this thesis, pedestrian motion is an important cue which describes pedestrian activity. Researchers utilize computer vision methods to detect and track individual pedestrians in videos, to build their motion trajectories. Pedestrian detection and tracking, in general, is a challenging problem. Difficulties in detection can arise due to changing

appearance patterns of both the pedestrian and the scene, pedestrian-like confusers (resembling pedestrian's shape), and low illumination conditions. Difficulties in tracking can arise due to sudden change in pedestrian motion, changing appearance patterns of both the pedestrian and the scene, nonrigid pedestrian structures, pedestrian-to-pedestrian and pedestrian-to-scene occlusions, and camera motion. Tracking is usually performed in the context of higher-level applications that require the location of the pedestrian in every frame. Typically, assumptions are made to constrain the tracking problem in the context of a particular application. The objective of this thesis is to understand pedestrian activity, which requires tracking of all pedestrians in the region. The work requires multi-pedestrian detection and tracking in medium crowd density (3 - 4 pedestrians per $m^2$). Techniques which cater to this requirement are discussed. Some of the widely used pedestrian detection and tracking methods reported in literature are briefly explained.

### 2.1.1 Pedestrian Detection

There is an extensive literature on object detection, but only relevant works on human detection [18, 19, 20, 21, 22] are highlighted here. Papageorgiou et al [18] described a pedestrian detector based on a polynomial Support Vector Machine (SVM) using rectified Haar wavelets as input descriptors, with a parts (sub window) based variant in [19]. Depoortere et al gave an optimized version of this approach in [23]. Gavrila & Philomen [24] took a more direct approach, extracting edge images and matching them to a set of learned exemplars using chamfer distance [25]. Such an approach has been used in a practical real-time pedestrian detection system [26]. Viola et al [20] built an efficient moving person detector, using AdaBoost meta algorithm [27] routine to train a chain of progressively more complex region rejection rules based on Haar-like wavelets and space-time differences. Ronfard et al [28] built an articulated body detector by incorporating SVM based limb classifiers over $1^{st}$ and $2^{nd}$ order Gaussian filters in a dynamic programming framework similar to those of Felzenszwalb & Huttenlocher [29] and Ioffe & Forsyth [30]. Mikolajczyk et al [21] used combinations of orientation position histograms with binary-thresholded gradient magnitudes to build a parts based method containing detectors for faces, heads, and front and side profiles of upper and lower body parts.

[31] provides a survey of various pedestrian detection techniques. Most of these methods are computationally expensive and hence only two techniques are explored in this thesis. They are Background Subtraction and Histogram of Oriented Gradients for Pedestrian Detection.

**Background subtraction** Background subtraction [32] is one of the oldest technique to detect moving objects, such as pedestrians [33, 34]. A robust background subtraction algorithm should be able to handle lighting changes, repetitive motions from clutter and long-term scene changes [35]. Several methods of background subtraction algorithm are reported in literature. In frame differencing method, pedestrians are detected by performing computations on the pixel intensities. Foreground masks, which represent potential pedestrians, are detected from video frames by subtracting the background $B(x, y, t)$ from the video frame $V(x, y, t)$ at time $t$. The background is assumed to be the frame at time $t$. This approach will only work for cases where all foreground pixels are moving and all background pixels are static [35] [36]. A threshold $Th$ on the pixel intensity is applied on this difference image to improve the subtraction results. Foreground masks which exceed a pre-defined pixel intensity threshold $Th$, are identified in [37]. An example of foreground generated using this approach is shown in Figure 2.2. In mean filter method [38], the mean of "$n$" previous frames is considered as the estimated background $B(x, y, t)$. In the method reported in [39], a Gaussian probabilistic density function (pdf) fitting is performed on the pixel intensity values of the most recent $N$ image frames. A running average (of every pixel intensity value) is computed to avoid fitting the pdf from scratch at every frame time. The centroid which is the 2-dimensional coordinate ($z_t = (x_t, y_t)^T$) of the identified foreground masks represents the spatial locations of the pedestrians. In [40], every pixel's intensity value in the video is modeled using a Gaussian mixture model. A simple heuristic determines which intensities are most probably of the background. Then the pixels which do not match these are called the foreground pixels. A complete survey of the various methods of background subtraction is available in [41].

When background subtraction is employed, stationary pedestrians become part of the background after some time (typically more than 10 seconds) when the background is updated periodically. Due to this limitation, Histogram of Oriented Gradients for

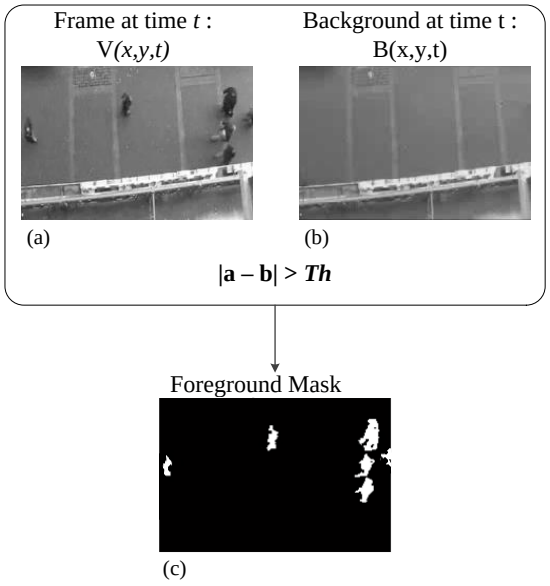



Figure 2.2: A foreground mask (c) generated after background subtraction and 'thresholding' $|a - b| > Th$.

Pedestrian Detection (will be called as HOG in this thesis) is explored in this thesis which can detect stationary pedestrians also. This method is briefly explained below.

**Histogram of Oriented Gradients for Pedestrian Detection (HOG)** HOG utilizes a Support Vector Machine (SVM) as a classifier to detect pedestrians. SVMs are supervised learning models that are used for classification and regression analysis. Typically, SVMs are used to classify data into two classes by computing the maximum margin hyperplane that differentiates one class from the other [42]. In the context of pedestrian detection, the two classes correspond to pedestrian class (positive samples) and non-pedestrian class (negative samples). Generally, SVMs require a large collection of samples from each class (i.e. positive samples and negative samples) for a better classification accuracy. In SVM, a decision model or decision function is created through the learning of labeled data set (see Figure 2.3). This decision model is then used to classify input data to pedestrian class or non-pedestrian class (Figure 2.4).

The steps in HOG are shown in Figure 2.5. In HOG, the input image frame is first converted to grey scale. Then, the grey scale image is decomposed into small connected

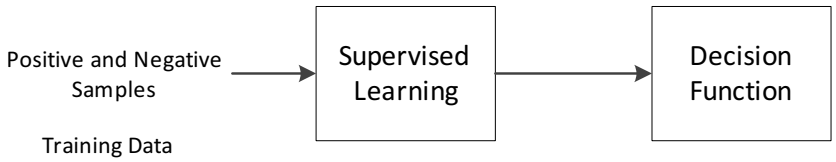



Figure 2.3: Building decision function of a Support Vector Machine (Supervised Learning Tool).

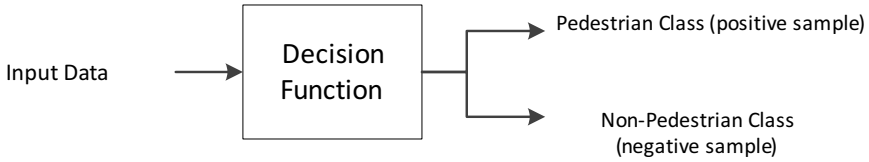



Figure 2.4: Classification using decision function.

regions (cells). Gradient vectors (image gradients) of each pixel in the cell is computed through measuring variation of pixel intensity values along x-direction and y-direction around each and every pixel. The gradient vector computation for every image pixel is explained with the help of Figure 2.6. The gradient vector for the black pixel is computed by subtracting the intensity values of the adjacent horizontal pixels and then dividing the difference by the difference of the adjacent vertical pixels. The gradient vector occurrences (which is the gradient orientation) are counted in each cell and the dominant gradient vector (which has the maximum number of occurrences in a cell) is selected as the representative of that cell. The Histogram of Oriented Gradients of the small connected regions is the collection of such representative gradient vectors [43]. The histograms are then labeled as positive (1 - a pedestrian) or negative (0 - not a pedestrian). These labeled histograms are the input to the SVM to build the SVM decision function. The built SVM decision function is then utilized to classify the new histograms. The histograms for two pedestrian images are shown in Figure 2.7. The concept behind the Histogram of Oriented Gradients is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. For improved accuracy, the histograms are contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination or

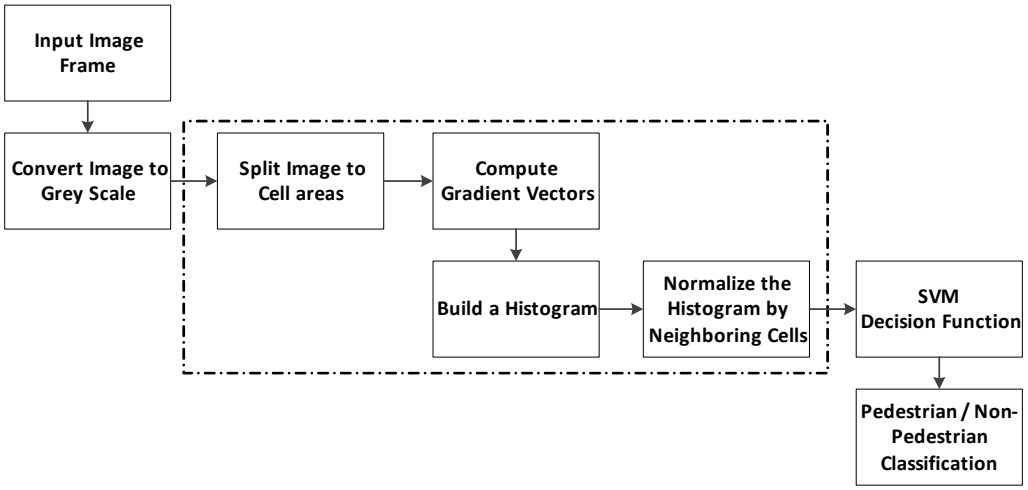



Figure 2.5: Steps in pedestrian detection using Histogram of Oriented Gradients.

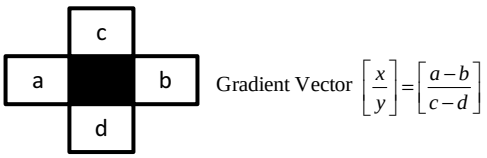



Figure 2.6: Computation of gradient vector for pixel highlighted in black.

shadowing. The HOG technique has a high pedestrian detection accuracy. It detects pedestrians from individual frames and does not need a background model for pedestrian detection. Both moving as well as stationary pedestrians can be detected by HOG. Detailed information about HOG is available in [1]. The pedestrian detection accuracy of HOG is discussed with examples in appendix A.

In this thesis, the centroid of the detected pedestrian represent the location of the pedestrian in the frame region.

### 2.1.2 Pedestrian Tracking

After a pedestrian has been detected, tracking is achieved by associating the centroids in the current frame to those in the previous video frame. The centroid of a pedestrian is represented as a state vector, $z_t = (x_t, y_t)^T$. Using Kalman filters, the centroid in the current frame is predicted ($\hat{z}_{t|t-1}$) from the estimation ($\hat{z}_{t-1|t-1}$) in the previous frame ,

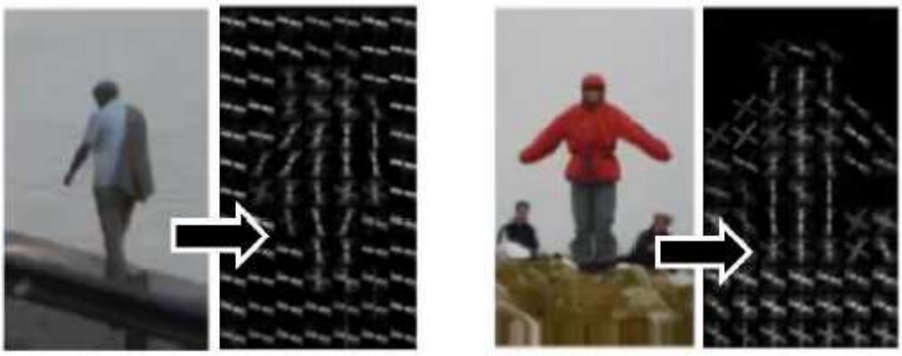


Figure 2.7: Pedestrian images and their Histogram of Oriented Gradients. Figure from paper by Dalal et al [1].

$\hat{z}_{t|t-1} = B_t \hat{z}_{t-1|t-1}$, where $B_t$ is the state transition model [44]. The updated posteriori centroid in the current frame is $\hat{z}_{t|t} = \hat{z}_{t|t-1} + K_t \tilde{e}_t$, where $K_t$ is the Kalman gain ([44], [45]) and $\tilde{e}_t = z_t - \hat{z}_{t|t-1}$.

Trajectories are sequences of flow vectors $F_i^t = \{f_1, ..., f_t\}$, where $f_t = [x_t, y_t, u_t, v_t]$ represents a pedestrian's (for example pedestrian i) motion at time $t$, represented by the centroid $z_t = (x_t, y_t)^T$, and $u_t$ and $v_t$ are the velocities in the $x$ and $y$ directions, respectively. An updated centroid in the current frame is matched with the trajectories (built till the previous frame) by solving the corresponding Linear Assignment Problem (LAP). The Hungarian algorithm [46] is used to solve the Linear Assignment Problem. There exist several other statistical data association techniques to tackle this problem. A detailed review of these techniques can be found in [47]. Matched trajectory-centroid pairs in the LAP solution are merged to extend the trajectory. Centroids that have no matching trajectory are used to start new trajectories. Trajectories which do not have a matching centroid are terminated. The coordinates in the trajectories are used to calculate the motion parameter variations (a time sequence of the distance from origin, velocity and direction of travel).

Apart from Kalman filters, there are other predictors (like Particle filter) which can predict and estimate the pedestrian position. Unlike the Kalman filter, the Particle filter uses unique Gaussian distributions for different pedestrians [48]. The pedestrians' motion are modeled by their Gaussians independently. If there is an abrupt change in the motion

of a pedestrian, it is unlikely to have an effect on other pedestrians' position estimations. In simple terms, if one of the pedestrians detected in the video starts to run, it is still possible for the system to consistently track the other pedestrians who are not running. In the case of the Kalman filter, the estimation of the next point for all the pedestrians will most likely be affected. However, the scenarios considered in this thesis does not involve abrupt change in the pedestrian motion.

## 2.2 Applications of Pedestrian Detection and Tracking

With the introduction of cameras for surveillance, pedestrian detection and tracking has become a pre-requisite for numerous surveillance applications. Intruder detection [49] is one of the standard applications of pedestrian detection and tracking. Pedestrians are detected and tracked to build their trajectories. Their trajectory coordinates are continuously checked to find if any pedestrian is located in an unauthorized region. The unauthorized region include airport runways, military zones, volcanic regions, nuclear power plants and many more. In public places, loiterers are detected by sensing the time a pedestrian spends in the place [50]. Pedestrian detection is also utilized to estimate crowd densities by counting the number of detections (termed as people counting [51, 52, 53]). Such applications aim to find the approximate level of crowd in a region of interest.

One of the specific and recent applications of people counting is counting people across pedestrian passage ways. Pedestrians entering and exiting a passage way are counted based on their direction of movement. Single [54] as well as multiple-camera [55] based people counting approaches are explored in the research literature. At high risk zones like country borders, efficient pedestrian detection and tracking techniques [56] are utilized to automatically focus the video camera on potential targets.

Pedestrian detection is a preprocessing step in facial recognition [57, 58] for some of the surveillance applications. In such applications, the pedestrians detected from the video frames are considered for processing in the facial recognition algorithms.

Pedestrian detection is an essential part in demographics analysis in commercial ar-

eas like shopping malls. In demographics analysis, the detected pedestrians are gender classified [59] and their age is estimated [60] to better understand the target audience of different products and shops. Here, gender classification and age estimation adopt vision based techniques, where either the facial features [61, 59] or the motion gait features [60] is utilized. Apart from demographic analysis, pedestrian stall occupancy determination [62] based on pedestrian traffic is a high demand application among shopping space developers. This application provides information on popularity of shops based on pedestrian traffic and pedestrian dwell time (time spent in the shop).

In the past decade, automotive researchers have turned their attention towards developing safety systems which could alert drivers of pedestrian traffic on roads. The system in [63] detects pedestrians using pedestrian shape models and tracks their movement to predict pedestrian movement onto roads from sidewalks. In [64], radar images are used instead of normal images to achieve the same objective.

This section outlined some of the basic applications of pedestrian detection and tracking. Motion trajectories which result from pedestrian detection and tracking are analyzed to generate higher level information like pedestrian behavior. Such applications are outlined in the next section.

## 2.3 Applications of Pedestrian Motion Trajectory Analysis

There are numerous applications of pedestrian motion trajectory analysis. Some of the state-of-the-art applications are outlined in this section. Behavior prediction interprets information by extracting sequences of actions (based on motion trajectories) of one or more persons using Hidden Markov Models [65] or Bayes Network [66] and also includes interpretation of statistical analysis of routine actions (for example the pedestrian browsing movement in shopping malls). Applications of identifying pedestrian crowds include identification of groups involving an arbitrary number of actors, such as identifying groups of people shopping together [67], locating queues at vending machines [68], identifying movement of groups of people [69], analyzing social interaction in small group conversations

[70], and detecting group formation and dispersal behaviors through statistical clustering of pairwise relational predicates [71].

Several methods to identify pedestrian groups are reported in the literature. In [72], pedestrians with similar velocities are grouped. In such an approach, pedestrians who are spatially far apart are falsely grouped together when they have a common velocity. A social group model based on the measurement of each individual's personal space is explored in [73]. Cupillard et al. [74] developed a tracker to parse and group individual trajectories of pedestrians walking together. The tracker detected moving regions to form trajectories of pedestrians, following the principle of Reid's Multi Hypothesis Tracking (MHT) [75]. The results are only shown on videos with several pedestrians (2-5) walking in a metro station, and the scalability of this method is unclear, considering the larger number of MHT hypotheses. Lau et al. [76] clustered 3D data points from a laser range finder into groups of human-sized blobs and adapted MHT to directly track moving, merging, and splitting groups over time. The theory of proxemics [16], proposed by anthropologist Edward Hall, defines groups based on ranges of personal and social interaction distances. In [77], Sochman uses social force models (SFM), to detect pedestrian groups by minimizing the SFM parameters. An algorithm in [5] identifies small pedestrian groups by recursive clustering (an agglomerative clustering approach) using the pedestrians' pairwise motion similarity. The algorithm does not identify large pedestrian groups due to a strict cluster termination criteria, which identifies a pedestrian to be a group member, if the pedestrian's motion is similar to at least half of the group members' motion. In [6], a similar recursive agglomerative clustering of pedestrians is performed based on pedestrian pairwise distance. The clustering algorithm does not consider the direction of travel to identify pedestrian groups and hence, falsely identifies pedestrians who are moving away from each other in close proximity as a pedestrian group. Data sets with several pedestrians are only used to validate the performance of the approach. The reported research works validate their performance using observed group ground truth and not the actual ground truth. The proposed Non-recursive Motion Similarity Clustering (NMSC) algorithm [78] (proposed in this thesis) considers distance, speed and direction of travel to identify pedestrian groups. The NMSC algorithm is validated using the actual group ground truth and it can identify large as well as small pedestrian groups with a simple

non-recursive clustering approach.

Inter-group and Intra-group behavior recognition [79, 80] are higher level applications, identifying interactions between the identified groups and thereby identifying a more complex form of information - Group dynamics. Group-Activity association [81, 82] identifies the action performers and the action performed based on motion trajectory information and the visual cues available (group identification based on image processing). The visual cues require an accurate training database which is difficult to create and needs to be recreated if new activities are included.

In this thesis, several algorithms are proposed to understand pedestrian activity. The thesis mainly focuses on utilizing the social psychological theory on pedestrian groups to identify pedestrian groups in video scenes. Several applications such as pedestrian meeting event identification, people counting at queue regions and crowd estimation at public locations (taxi queues, bus bays and train platforms) are discussed. These applications utilize the features identified from pedestrian group identification. The thesis also proposes methods to predict pedestrians' group (prediction of future pedestrian groups) and individual behavior (prediction of pedestrian approach to a destination) by utilizing the pedestrian motion parameter variations.

# Chapter 3

# Preliminaries

This chapter explains the steps to extract the two-dimensional coordinates of the pedestrians from the video clips. These are image processing steps. A pedestrian detector with a three-level blob filtering approach has been developed to improve the pedestrian detection accuracy. The three-level blob filter uses pedestrian detections from Background Subtraction and Histogram of Oriented Gradients for Human Detection. The filters attempt to eliminate some of the non-pedestrian confusers found in real world situations. The pin hole model to perform the translation of the three-dimensional coordinates to two-dimensional coordinates is explained.

## 3.1 Pedestrian Detection using Three-Level Blob Filter

In order to improve pedestrian detection accuracy, a pragmatic three-level blob filtering approach is developed and tested with videos from several test locations. The pedestrian motion trajectories extracted using this detection approach are used to detect social groups and to learn spatial distribution of pedestrians over a period of time. Several limitations of the existing pedestrian detection techniques [18, 19, 28, 32, 1] such as high computational complexity, pedestrian miss detections with partial body patterns and miss detections of stationary pedestrian are addressed by this approach.

### 3.1.1 Pedestrian Detector with Three-Level Blob Filtering

The pedestrian detection and tracking module adopted in this thesis employs the three-level blob filter approach. In this approach, two finite sets of blobs are detected from every video frame, using the Histogram of Oriented Gradients for human detection - Fast HOG[1][1] and Background subtraction (with 'Mixture of Gaussians' method [32, 33]). The false negatives in the Fast HOG detections which happen in the near camera region, false positives which arise due to illumination variations and the false positives which resemble the pedestrian body patterns are filtered out by the three levels of blob filtering. The three filters are used together with the Fast HOG and Background subtraction detections to achieve high pedestrian detection accuracy, for both indoor and outdoor locations, in day and night time.

Several test locations (Figure 3.1) in a residential hall at the National University of Singapore were utilized to test the pedestrian detection accuracy. These locations cover a wide range of pedestrian detection challenges. Camera locations D and E are different from the other camera locations because they are indoor locations with polished floors resulting in high reflections. These reflections are a source of false positives due to physical entities like object or shadows detected as pedestrians. Other critical challenges include varying illumination conditions, low illumination, high occlusion and pedestrian resembling confusers.

The three levels of blob filtering and the corresponding problems (false positives or false negatives) addressed are explained below.

**Level-1 Blob Filter** It is observed that Fast HOG performs better than Background Subtraction when complete body patterns are visible and may not detect pedestrians without complete body patterns. Fast HOG did not detect pedestrians whose complete body pattern is not available in the region near the camera (see Figure 3.2). Background Subtraction is able to detect the pedestrian in the near camera region, if he / she has even a slight movement. The level-1 blob filtering selects Background Subtraction detections

[1]This is a Graphics Processing Unit (GPU) implementation of Histogram of Oriented Gradients for human detection (HOG). Information about HOG is available in chapter (2) and real-time performance discussion is provided in appendix A.

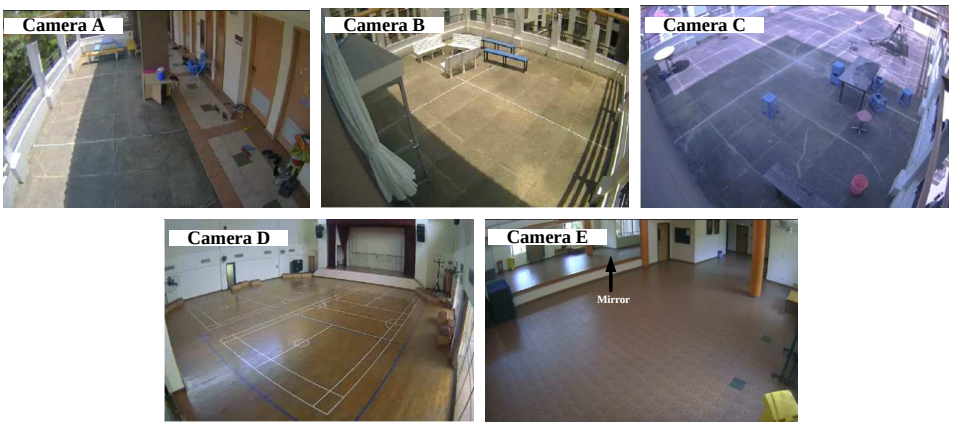



Figure 3.1: Various test locations selected for testing proposed algorithms.

($D_{i,B}$, where $i = 1...N$, $N$ is the number of blobs detected), which occur in the near camera region and Fast HOG detections ($D_{i,H}$) in the far camera region. The image region is manually divided into two: far camera region $R_f$ and near camera region $R_n$ (see Figure 3.3). Blobs of Fast HOG from $R_f$ and blobs of Background Subtraction from $R_n$ are selected according to (3.1) and (3.2). The set of selected blobs is passed on to the next level of blob filtering according to (3.3).

$$D_{i,H} \in D_H^1, if\ centroid(D_{i,H}) \in R_f \tag{3.1}$$

$$D_{i,B} \in D_B^1, if\ centroid(D_{i,B}) \in R_n \tag{3.2}$$

$$D_S^1 = D_H^1 \cup D_B^1 \tag{3.3}$$

The set of blobs $D_S^1$ is passed on to the level-2 blob filter.

**Level-2 Blob Filter** Pedestrian detection is also affected by sudden illumination variations which result in false positives. This might be due to a cloud pass or a partial shadow caused by moving entities obstructing the sunlight. Fast HOG is affected only if the illumination variation resembles a pedestrian while Background Subtraction is affected most of the time on a cloudy day when sudden illumination changes are likely to happen.

Figure 3.2: Pedestrian detection with Background Subtraction (left section) and Fast HOG (right section). Fast HOG misses the pedestrian as a complete body pattern is not available to detect a human.


Figure 3.3: Far $R_f$ and Near $R_n$ camera regions (manually marked) for camera D and E locations. Blobs of Fast HOG from $R_1$ and blobs of Background Subtraction from $R_2$ are selected by Level-1 Blob Filter.

Figure 3.4: Pedestrian detections without Level-2 blob filtering (first and third) and with Level-2 blob filtering (second and fourth). False positives are observed on the floor.


Figure 3.5: Fast HOG pedestrian detections without Level-3 blob filtering (first and third) and with Level-3 blob filtering (second and fourth). False positives are marked by red ovals.

To reduce false positives, pedestrian blobs' pixel areas are modeled as a Gaussian distribution. The Gaussian distribution's standard deviation $\sigma$ and mean $\mu$ are used to find the maximum ($b_{max} = \mu + \sigma$) and minimum ($b_{min} = \mu - \sigma$) blob area thresholds. Blobs with areas greater or lower than these thresholds are filtered out according to (3.4),

$$D^1_{i,S} \in D^2_S, if\ b_{min} \leq area(D^1_{i,S}) \leq b_{max} \tag{3.4}$$

where, $D^1_{i,S}$ denotes a blob from level-1 blob filter and $D^2_S$ is the set of blobs selected by level-2 blob filter.

The Gaussian modeling is repeated every time a substantial number of blobs are detected, to learn the new $b_{max}$ and $b_{min}$ values. This approach not only helps to remove false positives due to illumination variations but also makes the pedestrian detector intelligent in identifying the pedestrian blobs (even if shifted to a different zoom option - where pedestrian blobs' pixel areas are different). The set of blobs $D^2_S$ is passed on to the level-3 blob filter for further processing. Figure 3.4 shows the removal of the false positives when Level-2 blob filter is used.

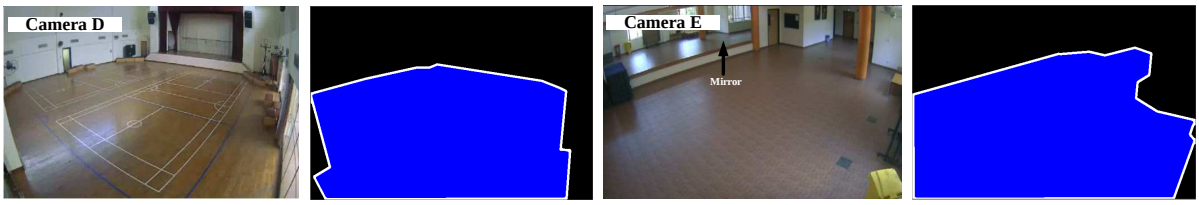



Figure 3.6: Masks $M$ (second and fourth) for Camera D (first) and Camera E (third) locations for Level-3 Blob filtering. A detected blob with it's lower edge partially or fully within the mask is selected by Level-3 Blob filter.

**Level-3 Blob Filter** The detected blobs which resemble the body pattern of a pedestrian are termed as pedestrian-like confusers. Fast HOG is affected by these confusers resulting in false positives. Spatially, these confusers are observed to happen on walls and on mirrors due to reflections (first and third picture in Figure 3.5). Blobs, which are not part of a floor region (a floor mask) are filtered out by the level-3 blob filter. In all the test locations (Figure 3.1), pedestrians are present on the floor. In this work, the floor mask $M$ is automatically detected using canny edge detector [83]. The canny edge detector is utilized to automatically detect closed contours and select the contour closest to the lower edge of the video frame for floor mask generation. Figure 3.6 shows floor masks generated for camera locations D and E. A Level-2 blob $D^2_{i,S}$ (where $i = 1...N$) is selected by the level-3 blob filter according to (3.5),

$$D^2_{i,S} \in D^3_S, if\ lower\ edge(D^2_{i,S}) \in M \tag{3.5}$$

Figure 3.5 (second and third picture) shows the removal of the pedestrian like confusers (false positives) on walls and mirrors after level-3 blob filtering. Blobs ($D^2_{i,S}$) whose lower edges are not within the floor mask $M$, are filtered out. The $D^3_S$ blobs are processed by the tracking module. The detected pedestrians are tracked using Kalman filters [44] and the Hungarian algorithm [46] is used to build the pedestrians' trajectories.

### 3.1.2 Performance Evaluation and Discussion

The pedestrian detection accuracy of the three-level blob filtering approach is discussed in terms of the precision and recall scores [84]. Precision and recall scores range from 0 to

1, where scores close to 1 indicate good performance of the pedestrian detector. A short video clip of 20 minutes is selected from recordings of all the camera locations (Camera A, B, C, D and E) for the evaluation. Precision and recall scores are calculated according to (3.6) and (3.7).

$$Precision\, P = \frac{t_p}{t_p + f_p} \tag{3.6}$$

$$Recall\, R = \frac{t_p}{t_p + f_n} \tag{3.7}$$

where, a pedestrian detected is a true positive $t_p$, a pedestrian not detected is a false negative $f_n$ and a detection which is not a pedestrian is a false positive $f_p$.

Precision and recall scores are calculated by manual counting and confirming the detections against the ground truth. Table 3.1 reveals that the three-level blob filtering approach performs better than the individual Background Subtraction and the Fast HOG techniques as the precision and recall scores are higher for this approach. A video clip of 20 minutes is selected for different camera locations (Cameras A, B, C, D and E) for this evaluation. Results for cameras C and D have a lower recall rate due to some pedestrians remaining stationary in the near camera region, where only the Background subtraction detections are considered. The stationary pedestrians become part of the background environment and are no longer detected by Background subtraction.

A performance comparison is carried out using standard research data sets such as the PETS CAVIAR, ETH BIWI walking pedestrian and the INRIA data sets, against standard detection techniques like HOG, C4 and background subtraction.

The ETH BIWI walking pedestrian [2], PETS-ECCV'2004 - CAVIAR [3] and the INRIA [4] video data sets are used for the evaluation. These video data sets record different motion configurations such as pedestrians walking alone and meeting with others. Snapshot from these data sets are shown in Figure 3.7.

Table 3.2 reveals that the three-level blob filtering approach performs on par or better than the other standard pedestrian detection techniques tested in research video data sets. We come to this conclusion based on the higher precision and recall scores three-level blob filter compared to the other techniques.

Table 3.1: Precision $P$ and Recall $R$ scores for Background subtraction, Fast HOG and Three-Level Blob Filtering approach.

| Camera location | Background Subtraction | | Fast HOG | | Three-Level Blob Filter | |
|---|---|---|---|---|---|---|
| | P | R | P | R | P | R |
| A | 0.78 | 0.77 | 0.63 | 0.68 | 0.79 | 0.80 |
| B | 0.65 | 0.67 | 0.69 | 0.71 | 0.72 | 0.82 |
| C | 0.64 | 0.69 | 0.62 | 0.73 | 0.71 | 0.83 |
| D | 0.43 | 0.49 | 0.52 | 0.57 | 0.68 | 0.62 |
| E | 0.41 | 0.56 | 0.54 | 0.61 | 0.74 | 0.75 |

Figure 3.7: Snapshots from the BIWI Walking Pedestrians [2] (left side), the PETS-ECCV'2004 - CAVIAR [3] (middle) and the INRIA [4] (left side) video data sets.

Table 3.2: Precision $P$ and Recall $R$ scores for Background subtraction, Fast HOG, C4 detector and Three-Level Blob Filtering approach with standard research video data sets.

| Data sets | Background Subtraction | | Fast HOG | | C4 Detector | | Three-Level Blob Filter | |
|---|---|---|---|---|---|---|---|---|
| | P | R | P | R | P | R | P | R |
| PETS CAVIAR[3] | 0.65 | 0.61 | 0.68 | 0.67 | 0.75 | 0.76 | 0.74 | 0.78 |
| ETH BIWI walking pedestrian[2] | 0.69 | 0.68 | 0.67 | 0.62 | 0.68 | 0.80 | 0.70 | 0.79 |
| INRIA [4] | 0.57 | 0.58 | 0.74 | 0.69 | 0.69 | 0.80 | 0.73 | 0.85 |

## 3.2 Real-world to Image Plane Coordinates Translation

The algorithms proposed in this thesis process the two-dimensional image pixel coordinates. The translation from the three-dimensional real-world coordinates to the two-dimensional image plane coordinates is performed using the pin hole model of cameras. This model explains the 3D to 2D distance translation carried out.

In this model, a scene view is formed by projecting 3D points into the image plane using a perspective transformation.

$$s.m' = A[R|t]M'$$

or

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

Where, $(X, Y, Z)$ are the coordinates of a 3D point in the world coordinate space, $(u, v)$ are the coordinates of the projection point in pixels, $A$ is a camera matrix, or a matrix

of intrinsic parameter, $(c_x, c_y)$ is a principal point that is usually at the image center and $(f_x, f_y)$ are the focal lengths expressed in pixel units. Thus, if an image from the camera is scaled by a factor, all of these parameters should be scaled (multiplied/divided, respectively) by the same factor. The matrix of intrinsic parameters does not depend on the scene viewed. So, once estimated, it can be re-used as long as the focal length is fixed (in case of zoom lens). The joint rotation-translation matrix $[R|t]$ is called a matrix of extrinsic parameters. It is used to describe the camera motion around a static scene, or vice versa, rigid motion of an object in front of a still camera. That is, $[R|t]$ translates coordinates of a point $(X, Y, Z)$ to a coordinate system, fixed with respect to the camera. The transformation above is equivalent to the following (when $Z \neq 0$):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$u = f_x * x' + c_x$$

$$v = f_y * y' + c_y$$

## 3.3 Summary

A pedestrian detector with a three-level blob filtering approach has been developed. These filters attempt to eliminate some of the non-pedestrian confusers found in real world situations. Several pedestrian detection issues in indoor, outdoor locations, at low light, high illumination variation, pedestrian shadows, partial body patterns, pedestrian like confusers on walls and mirror reflections, were discussed and addressed. Experimental results show that the pedestrian detector has a better pedestrian detection accuracy

compared to the individual Fast HOG and Background subtraction techniques. The translation of the three dimensional coordinates to two dimensional coordinates is explained using the pin hole camera model.

**Acknowledgment** The author would like to thank the NUS Sheares Hall for providing access to the video footage for the experiments and the NUS Ambient Intelligence (AMI) lab in Interactive Digital Media Institute (IDMI) for their support in carrying out this research work.

# Chapter 4

# Pedestrian Group Identification

In order to understand the pedestrian activities in public locations, it is necessary to understand the pedestrian crowds. In public domains, one can think of numerous applications derived from identifying pedestrian crowds in a video scene. Pedestrian crowds are often found in transportation hubs such as bus and rail interchanges or in shopping malls. Pedestrian crowds can be analyzed by tracking the movement of pedestrians. By identifying and analyzing the underlying social groups which are formed over a period of time, crowd behavior can be better understood. Automatic group identification and tracking can therefore highlight regions of interest from numerous video feeds, thereby aiding the security personnel in the surveillance of the area.

In this chapter, an automatic method to identify social groups of pedestrians is discussed. Inspired by social psychological principles on group behavior [17], this method utilizes a novel Non-recursive Motion Similarity based Clustering (NMSC) algorithm to cluster pedestrians based on their motion similarities. Individual pedestrians are detected and tracked in video scenes. Pedestrians are automatically clustered based on their pairwise motion similarity by considering their relative distance, relative speed and relative direction of motion. Pedestrian clusters which persist for a period of time are identified as social groups. The theories on pedestrian group behavior are explained in the next section.

## 4.1 Pedestrian Group Behavior Theories

Pedestrian motion in a crowd is influenced by social psychological factors such as culture and personal space. For example, people in Latin America tend to move closer to each other while people in North America have larger personal spaces [85]. Irrespective of the nationality, a group of friends or a family walking together tend to maintain close proximity among them. Such prolonged proximity exhibits group behavior among pedestrians. There are two major theories on group behavior. The first considers the entire crowd as a single entity. Scholars have assumed that crowds transform individuals so that the resulting group begins to exhibit a homogeneous "group mind" that is highly emotional and irrational [85]. The second treats everyone as independent members acting to maximize their own utility [15]. Some researchers use a combination of the above two theories to model group behavior. For example, certain research works assume that the crowds are composed of groups, formed by individuals having relations to one another, making them interdependent to some significant degree [86]. Studies by McPhail [87] and Johnson [88] highlights this form of group behavior in crowds. The proposed method to identify social groups is based on this third approach. The method is discussed in the next section.

## 4.2 Automatic Pedestrian Group Identification

According to social psychological research [17], pedestrians are likely to be a social group if they are separated by a distance of less than 2 m, traveling almost at the same speed (relative speeds less than 0.4 m/sec) in almost the same direction (less than 3 degree of difference) for more than a few seconds (typically more than 3 seconds). These parameter thresholds are found to be consistent with social groups in places with low crowd density($<$ 2 pedestrians per $m^2$) to moderate crowd density (3-4 pedestrians per $m^2$) with no non-human factors (such as an emergency in natural calamity) to influence the pedestrian motion.

The stages to identify social groups of pedestrians in a video scene are explained below. The stages include pedestrian tracklet extraction, pedestrian clustering in the

NMSC algorithm, and social group identification. Figure 4.1 provides a brief explanation of the stages in the pedestrian group identification.

### 4.2.1 Tracklet Extraction

Pedestrians are detected utilizing the Three-Level Blob Filtering approach explained in Chapter 3. The centroid ($z_i = (x_i, y_i)$) of a detected pedestrian represents the location of the pedestrian. The detected pedestrians are tracked using Kalman filters and the Hungarian algorithm is used to build the pedestrians' trajectories. Trajectories are sequences of flow vectors $F_i^t = \{f_i^1, ..., f_i^t\}$, where $f_i^t = [x_i^t, y_i^t, u_i^t, v_i^t]$ is the $i^{th}$ pedestrian's motion at time $t$ (represented by the centroid $z_i^t = (x_i^t, y_i^t)$), and $u_i^t$ and $v_i^t$ are the speed components in the $x$ and $y$ directions, respectively. The coordinates in the trajectories are used to calculate the motion parameters (time sequences of distance from origin and velocity of travel). These motion parameters are used to cluster the pedestrians.

In this method, a moving window, also known as a tracklet $F_i^{T_i}$ (for pedestrian $i$ with trajectory length $T_i$), is utilized to select the coordinates from the latest thirty video frames for clustering. Pedestrians may split and form new groups within short time durations. Therefore, if a pedestrian's entire trajectory[1] information is considered for motion similarity calculation, the recent motion similarities are suppressed and the new social groups (which may exist only for a short time) may not be identified. Groups which exist for a short time duration are observed in video data sets in [2, 3], and the video data set (Sec. 4.3.1) recorded at the National University of Singapore.

### 4.2.2 Non-recursive Motion Similarity Clustering (NMSC) Algorithm to Cluster Pedestrians

For pedestrian clustering, pedestrians are compared in pairs, using their tracklets which are used to calculate the pedestrians' motion parameters. The tracklets of two pedestrians are considered similar, if their corresponding motion parameter values satisfy

[1]Entire trajectory consists of the pedestrian coordinates from all the video frames since the time the pedestrian entered the scene.

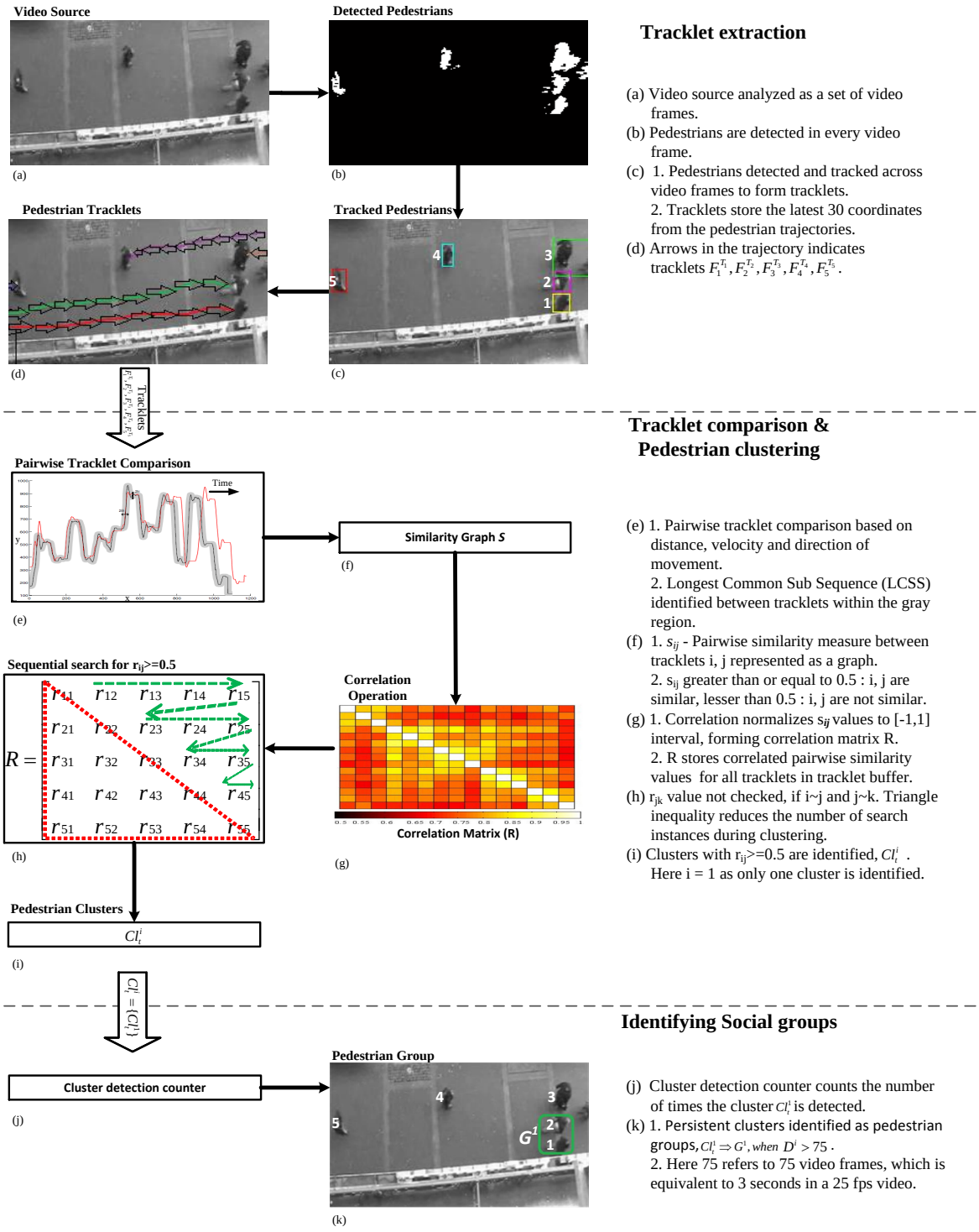



Figure 4.1: Stages in automatic pedestrian group identification. Identified pedestrian group highlighted with a green box.

Table 4.1: Important parameters in NMSC clustering algorithm

| Parameter | Video resolution | |
|---|---|---|
| | 112×160 | 176×220 |
| Euclidean distance threshold ($s_\epsilon$) (pixels) | 20 | 35 |
| Relative velocity difference threshold ($V_\epsilon$) (pixels per iteration) | 1 | 1 |
| Similarity measure controller ($\sigma$) (no unit) | 0.2 | 0.2 |
| Motion similarity threshold ($\triangle$) (no unit) | 0.5 | 0.5 |

the following criteria which are adopted from social psychological findings on pedestrian groups [17]:

- distance between pedestrians $< s_\epsilon = 2$ m
- relative speed difference $< v_\epsilon = 0.4$ m/s
- relative difference in the direction of travel $< d_\epsilon = 3$ deg.

The above motion parameter threshold values are converted to equivalent pixel values. The clustering algorithm works based on the pixel values of these motion parameter thresholds. Instead of considering the speed and direction differences separately, as in [12], a velocity vector, $V_{i,T_i}$ is formed from the speed and direction of travel. The infinity norm of the velocity difference $\left(\left\|V_{i,T_i} - V_{j,T_j}\right\|_\infty\right)$ between two pedestrians ($i$, $j$) is used as one of the similarity measure as it is more robust against noise. The corresponding relative velocity threshold is represented as $V_\epsilon$. The equivalent pixel values for the motion parameter thresholds for distance and relative velocity between pedestrians are given in Table 4.1.

The Longest Common Sub Sequence (LCSS) tool is used to search for matching centroid points between pedestrian tracklets, that are within a small euclidean distance $s_\epsilon$ within a time window $\delta$ (selected as 10 consecutive centroid points in the tracklet). The LCSS is robust to noise and outliers, as not all points in a tracklet need to be matched. A variant of LCSS [89, 90] is adopted in this work,

$$D_{LCSS}(F_i^{T_i}, F_j^{T_j}) = \frac{LCSS(F_i^{T_i}, F_j^{T_j})}{\min(T_i, T_j)}, \tag{4.1}$$

where, $LCSS(F_i^{T_i}, F_j^{T_j})$ stores the similarity count, which is the number of pairwise matching points between two pedestrians' tracklets defined by $F_i^{T_i}$ and $F_j^{T_j}$ of corresponding length, $T_i$ and $T_j$. The recursive LCSS definition is formulated in (4.2),

$$LCSS(F_i^{T_i}, F_j^{T_j}) = \begin{cases} 0, & T_i = 0 | T_j = 0, \\ 1 + LCSS(F_i^{T_i-1}, F_j^{T_j-1}), & d_E(z_{i,T_i}, z_{j,T_j}) < s_\epsilon \\ & \& \left\| V_{i,T_i} - V_{j,T_j} \right\|_\infty < V_\epsilon \\ & \& |T_i - T_j| < \delta \\ \max(LCSS(F_i^{T_i-1}, F_j^{T_j}), & \\ LCSS(F_i^{T_i}, F_j^{T_j-1})), & otherwise, \end{cases} \tag{4.2}$$

LCSS identifies two centroid points $(z_{i,T_i}, z_{j,T_j})$ as a match, only when the conditions in (4.2) is satisfied.

The pairwise similarity between pedestrians is modeled using graph theory. A similarity graph $S = \{s_{ij}\}$ is built with the similarity count calculated by LCSS in (4.2). A Gaussian kernel function is used to construct the similarity graph $S$. The similarity values, $s_{ij}$, are given by,

$$s_{ij} = e^{D^2_{LCSS}(F_i^{T_i}, F_j^{T_i})/2\sigma^2} \in [0, 1], \tag{4.3}$$

where, $\sigma$ describes a tracklet neighborhood. $\sigma$ is chosen as 0.2 and results show that it worked well in this algorithm.

A correlation matrix, $R = [r_{ij}]$, is formed from the similarity matrix, $S$ as in (4.4),

$$C = SS^T, \quad r_{ij} = c_{ij}/\sqrt{c_{ii}c_{jj}}, \quad -1 \leq r_{ij} \leq 1 \tag{4.4}$$

where, $c_{ij}$ and $r_{ij}$ are the elements of $C$ and $R$ respectively. Using $R$, pedestrian clustering is performed as a sequential search for correlation values of at least $\triangle = 0.5$. Pedestrians $i$ and $j$ are clustered, if $r_{ij} \geq 0.5$. Otherwise, $i$ and $j$ remains as individuals. The search (Figure 4.2) is performed on the elements of $R$ above the diagonal elements

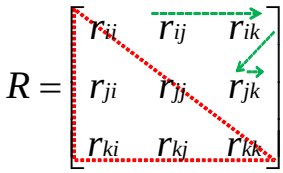



Figure 4.2: Search pattern for clustering using correlation matrix R for three pedestrians $i$, $j$, $k$. Search direction is along the dotted green line with arrow.

because this matrix is symmetric and the diagonal elements are not valid comparisons as they are auto-correlation values of each pedestrian. The sequential search demonstrates the use of the triangle inequality property [91]. For example, three pedestrians, $i$, $j$ and $k$, are clustered together if $r_{ij} \geq 0.5$ and $r_{ik} \geq 0.5$, regardless of whether $r_{jk} \geq 0.5$. This is because pedestrian $i$ is common to both $j$ and $k$. The search operation does not check $r_{jk}$, and this helps to reduce execution time in clustering. This clustering algorithm is applied to every video frame. The identified clusters are represented as

$$Cl_t^i, \quad i = 1, \ldots, N \tag{4.5}$$

where, $N$ is the total number of clusters which exist at time, $t$. The clusters are tracked across all video frames.

The proposed clustering approach is non-recursive with only one similarity matrix calculation with a single clustering iteration per video frame. While, the existing pedestrian clustering algorithms [5, 6] have numerous similarity matrix calculations with a recursive clustering approach (where clustering is performed numerous times per video frame until a clustering termination criteria is satisfied, an agglomerative clustering). Hence, the proposed clustering approach has a lower execution time than that of [5, 6]. The sequential search adopted in the proposed clustering approach reduces the execution time further as it does not search across all correlated pairwise scores (as explained in the above 'three pedestrians' example). While in [5, 6], all pairwise scores are checked at every clustering iteration.

### 4.2.3 Social Group Identification

The persistent clusters are identified as social groups. The condition to declare a pedestrian cluster $Cl_t^i$ as a social group $G^i$ is stated in (4.6).

$$Cl_t^i \Rightarrow G^i, if\ D^i > 75 \tag{4.6}$$

where, $D^i$ is the detection counter, storing the number of times $Cl_t^i$ was detected. The value of the detection counter $D^i$ is incremented by a value of one every time (every video frame) cluster $Cl_t^i$ is detected. When the value of $D^i$ is greater than 75 for a 25 frames per second video, cluster $Cl_t^i$ is identified as a group. This is equivalent to checking whether the pedestrian cluster $Cl_t^i$ persisted for 3 seconds (minimum persistence time threshold derived from the social group definition [17, 12]). In simple terms, pedestrian clusters which persisted for more than three seconds are identified as social groups. Otherwise, the cluster members are declared as individuals who happen to be walking momentarily in close proximity to one another.

The NMSC algorithm is called only when more than two pedestrians are in a scene and the pedestrians are detected in at least three consecutive video frames. The proposed NMSC algorithm is evaluated for group identification accuracy, real-time performance and computational complexity in the next section.

## 4.3 Performance Evaluation and Discussion

Several aspects of the NMSC algorithm are evaluated, namely: performance against human observer's judgment using the Kappa score measurement [92], group identification accuracy when compared to that of the existing pedestrian grouping algorithms [5, 6], computational complexity and real-time performance. The pedestrian data sets used for the clustering algorithm evaluation are first explained in Section 4.3.1 below.

### 4.3.1 Pedestrian Data sets

Video data sets which record pedestrians performing different motion configurations such as forming groups and moving along passages are used for evaluation purposes.

Figure 4.3: Snapshots from the BIWI Walking Pedestrians [2] (left side) and the PETS-ECCV'2004 - CAVIAR [3] (right side) video data sets.

Synthetic data sets are also created to test the pedestrian clustering algorithms.

**Benchmark Video Data sets** The PETS-ECCV'2004 - CAVIAR [3] and the BIWI Walking Pedestrians [2] video data sets (in Figure 4.3) are used for the evaluation. These video data sets record different motion configurations such as pedestrians walking alone and meeting with others. The standard research video data sets are of short duration and do not have information on the actual pedestrian groups which exists in the region. A human observer was employed to annotate the video and record the ground truth of pedestrian groups. Another video data set was created where the ground truth was directly recorded from the pedestrians. Such an approach eliminates the human judgment errors which affect the quality of the group ground truth. This data set is explained in the next section.

**NUS Movie Event (NUSME) Data set** Most of the available data sets in the literature [2, 3] do not record the ground truth for group formation and movement. The data set in [5] recorded the ground truth for group movement but the video is not available for analysis. Hence, a new video data set with group ground truth was created for the purpose of testing the proposed algorithms.

The data set records the pedestrian motion in the lobby area of a movie screening event at the National University of Singapore. The event attracted more than 200 visitors in 118 unique pedestrian groups, out of which 20 groups had their members occluded during some time of their existence.

The region was video recorded from an aerial camera view point. Figure 4.4 shows the

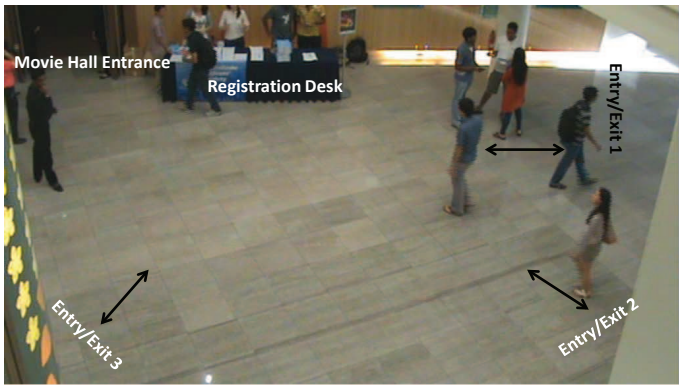



Figure 4.4: Video frame region in NUSME data set.

region with the location of the registration desk. The region has three points of entry / exit. In order to build ground truth without error of judgement, the visitors were requested to fill a form with the number of people in their group, time of visit and identification mark like shirt color. The recorded ground truth has information on the size of the group and indicators to locate the visitors in the video. A sample of the ground truth recorded is shown in Table 4.2. Numerous groups which existed for short durations (3 seconds and less) were observed in the movie event. Groups which have a large distance (greater than the social psychological distance threshold of 2 m) of separation between pedestrians were observed and there were instances where individuals walk together (Figure 4.5) even though they do not form a group.

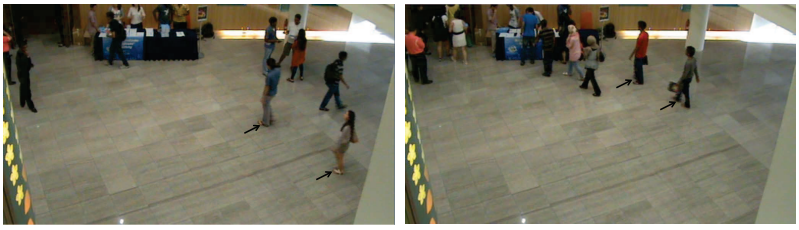


Figure 4.5: Group members walking far apart (left). Individuals walking close (right).

**Synthetic data sets** The synthetic data sets simulate motion configurations of pedestrian movements, group formation and deformation. The data sets were recorded using a mouse tracking algorithm, where each continuous mouse movement mimics a pedes-

Table 4.2: Sample of ground truth form used to record NUSME Data set

| Video Name | NUSME Video | | |
|---|---|---|---|
| Video Time | Time of Visit | Identification mark | Number of people |
| 0.00 | 5.17 pm | head scarf | 3 |
| 0.06 | 5.17 pm | red skirt | 2 |
| 0.16 | 5.17 pm | orange skirt | 2 |
| 0.58 | 5.18 pm | white T-shirt | 1 |
| 2.43 | 5.20 pm | guy with mobile | 2 |
| 2.51 | 5.20 pm | checked top | 1 |
| 2.52 | 5.20 pm | white top | 2 |
| 3.42 | 5.21 pm | black skirt | 1 |

trian movement. Different crowd configurations are simulated with 2, 4, 8, 16 and 32 pedestrians in the video frame region.

Motion configurations in the synthetic data sets are shown visually in Figure 4.6. Each pedestrian is coded by an unique color for the trajectory. Synthetic data sets are created to check the clustering algorithm's performance with different motion configurations occurring simultaneously and at different crowd densities. The highlights of the synthetic data sets are as follows - data set 1: group splits after some time, data set 2: pedestrians leave and join a group, data set 3: groups with different directions of travel and different speeds of movement, data set 4: groups with different directions of travel and data set 5: groups with different directions of travel and different speeds of movement. Combinations of these data sets, having 16, 32 pedestrians, were also used for performance evaluation.

Table 4.3 lists the motion configurations and the clustering algorithm's features analyzed using the data sets. The walking together action is created to check whether the pedestrian clustering algorithm is able to detect and track the group. Pedestrians moving in different directions from other pedestrian groups is simulated (in synthetic data sets 3, 4 and 5) to check how the directional information affects the identification of social groups. The performance evaluation is discussed in the following sections.

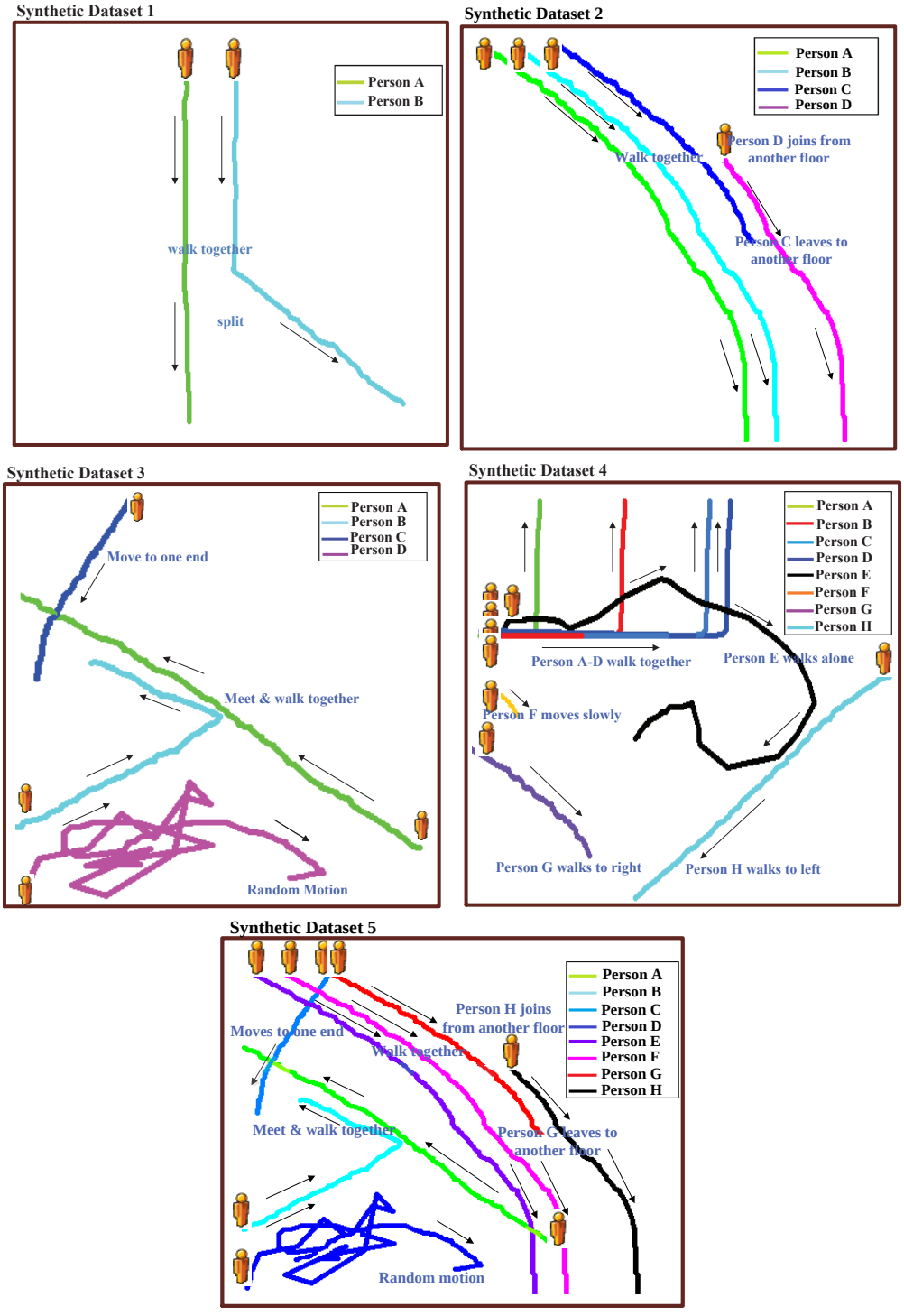



Figure 4.6: Visualization of synthetic data sets.

Table 4.3: Group Information in synthetic data sets, each motion configuration (in second row) identified corresponds to a feature analyzed (in third row)

| Data set Information | Synthetic Data set | | | | |
|---|---|---|---|---|---|
| | Data set 1 | Data set 2 | Data set 3 | Data set 4 | Data set 5 |
| Number of pedestrians | 2 | 4 | 4 | 8 | 8 |
| Motion configurations | 1) walk together and split | 1) walk together & join, 2) person leaves | 1) meet & walk together, 2) move opposite direction to a group, 3) random motion | 1) walk together, 2) walk at different speed to a group (person E), 3) walk together at different speed (person F, G), 4) Walk close to a person momentarily (Person H). | Combination of synthetic data set 2 & 3. |
| Feature analyzed | 1) group identification and execution time. | 1) group identification. 2) response of clustering algorithm to change in group and execution time. | 1) group identification. 2) role of direction of travel in detecting groups. 3) response of clustering algorithm to momentary similar pedestrians and execution time. | 1) group identification. 2) & 3) role of velocity of travel in detecting groups. 4) Effect of time period to detect a pedestrian group from a pedestrian cluster and execution time. | As in synthetic data set 2 & 3 and execution time. |

Table 4.4: Kappa scores for various data sets which were tested using the NMSC algorithm

| **Data set** | **Video 1 [3]** | **Video 2 [2]** | **Synthetic 1** | **Synthetic 2** | **Synthetic 3** | **Synthetic 4** | **Synthetic 5** |
|---|---|---|---|---|---|---|---|
| **Kappa score** | 0.72 | 0.67 | 1 | 0.96 | 0.64 | 0.62 | 0.85 |
| **Group sizes observed** | two | two | two | three & four | four | two | two, three & four |

### 4.3.2 Kappa Score Measurement

The Kappa score [92] measures the degree of agreement between the group identification results and the group ground truth. The group ground truth of individuals and groups is annotated by four human observers, in the synthetic and video data sets. The Kappa $\kappa$ score for a group size $g$ is defined as,

$$\kappa_g = \frac{P_0 - P_c}{1 - P_c}, \tag{4.7}$$

where, $P_0$ is the relative observed agreement among the observers, and $P_c$ is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly classifying a group size. If the observers are in complete agreement, then $\kappa_g = 1$. If there is no agreement among the observers other than what would be expected by chance (as defined by $P_c$), $\kappa_g = 0$. A Kappa score close to 1 indicates good performance of the clustering algorithm as there is a high degree of agreement between the ground truth and the group identification results.

Table 4.4 reveals considerable agreement (0.78 - an average of all the Kappa scores) between the ground truth agreed by all observers and the group detection results, indicating good performance of the NMSC algorithm. The high degree of agreement indicates that the proposed NMSC algorithm detects groups like a human observer most of the times.

### 4.3.3 Group Identification Accuracy

Visual aids are used to compare the group identification results of the proposed NMSC and the existing algorithms [5, 6]. The grouping results for the synthetic data set 5 is demonstrated in Figure 4.7, which is considered to be the most complicated among all the data sets used for the evaluation. To aid in understanding the evolution of the groups, pedestrian motion trajectories are marked with blue boxes highlighting the identified groups. The blue boxes are not erased with time. This representation is termed as the Temporal Grouping Plot (TGP). Subplots 2 to 4 in Figure 4.7 are TGPs.

From the TGPs in Figure 4.7, it is clear that the ETH Flock detection approach in [6] identifies pedestrians as groups even when they travel in different directions but close to each other because the adopted trajectory similarity measure does not consider direction of travel for clustering. The clustering algorithm in [5] cannot identify groups of size larger than 3 and bigger intra-group distance (i.e. sparse groups[2]) because of the strict cluster termination criteria. As explained in [5], only compact groups (which are well connected) are identified by the graph based clustering termination criteria. The termination criteria ensures clustering of a pedestrian to a group only when the pedestrian is related (similar in motion) to at least half of the other pedestrians in the group. Pedestrians who do not satisfy this criteria remain as individuals. The proposed NMSC algorithm can identify groups with large intra-group distances (more than 1 meter - sparse groups) with appropriate setting of the motion parameter thresholds. This is because the triangle inequality property in NMSC does not look for complete connectivity between group members, all the members need not be connected to each other but at least one connection should exist between the group members.

### 4.3.4 Computational Complexity

The computational complexity of the pedestrian grouping techniques are compared by analyzing their computational orders. The overall order is determined from the order of

[2]In sparse groups, pedestrian group members do not walk close to each other. Some members might have a physical separation of more than two meters from the other. This is a common type of social group motion, for example, five pedestrians walk side by side. The first pedestrian might be separated by more than 1 meter in distance from the fifth pedestrian.

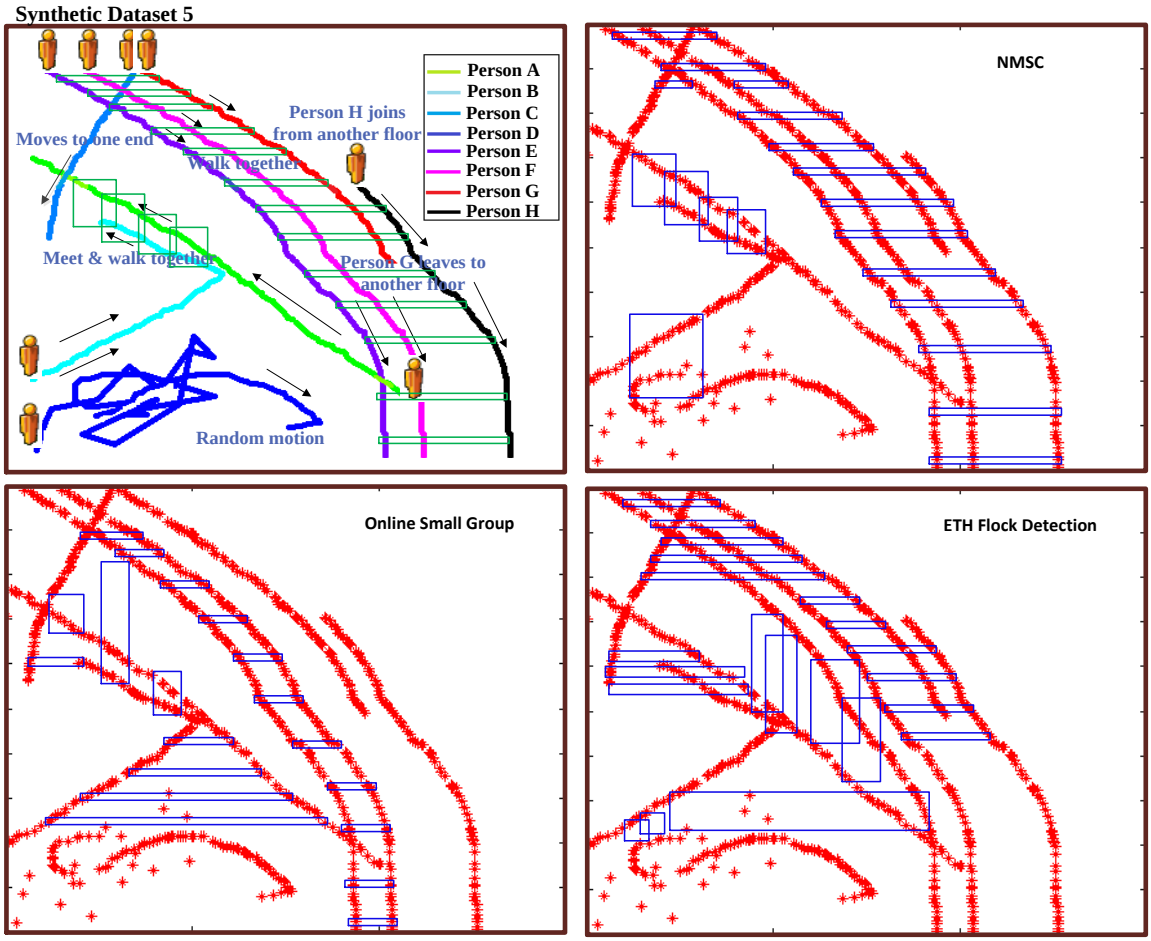



Figure 4.7: Synthetic data set 5 visualization (top left). Temporal Grouping Plot (TGP) for the NMSC algorithm (top right), for Online small group detection [5] (bottom left) and for ETH Flock Detection [6] (bottom right) algorithms. The pedestrian groups are marked by green (ground truth) and blue boxes (group identification results).

the individual stages of the grouping techniques.

In the proposed grouping technique, the order for:

- Similarity graph formulation is $Tn(n-1)$, where $n$ is the number of trajectories compared in a trajectory buffer of "$T$" coordinate storage locations,
- Correlation operation on the similarity graph is $\frac{n}{2}(n-1)$,
- Sequential search operation in the NMSC algorithm is $\frac{n}{2}(n-1)$.

The maximum among the various stages is $O(n^2)$, which is the order of the proposed grouping technique.

For the grouping technique in [5], the order for:

- Similarity graph formulation is $Tn(n-1)$,
- Cluster termination check is $2A(A-n)$, where $n$ is the number of trajectories and "$A$" is the group size,
- Agglomerative clustering is $n^3$.

The order of the framework is therefore $O(n^3)$. Similarly, the order of the grouping technique in [6] is found to be $O(n^3)$. The proposed NMSC algorithm is the simplest among the compared algorithms.

### 4.3.5 Real-time Performance - Execution Time Comparison

The execution time of the clustering algorithms are recorded. Tables 4.5 (for synthetic data sets) and 4.6 (for video data sets) list the mean execution time (in millisecond) for the clustering algorithms. The grouping techniques are evaluated in a computer powered by an Intel i7 processor at 3.4 GHz processing speed. Tables 4.5 and 4.6 reveal that the proposed NMSC algorithm performs at least an order of magnitude faster than the best of the other two clustering algorithms [5, 6]. Table 4.5 reveals that the other algorithm's execution times double with each increased crowd density, while the NMSC algorithm's execution time does not increase considerably. This is consistent with the lower computational

Table 4.5: Mean execution time at different crowd densities, using synthetic data sets

| **Number of pedestrians** | **Clustering Algorithms** | | |
|---|---|---|---|
| | **NMSC (proposed)** | **Online Small Group [5]** | **ETH Flock Detection [6]** |
| 4 | 0.02 ms | 1.96 ms | 0.71 ms |
| 8 | 0.02 ms | 3.33 ms | 1.22 ms |
| 16 | 0.04 ms | 7.23 ms | 2.32 ms |
| 32 | 0.05 ms | 15.33 ms | 5.26 ms |

Table 4.6: Mean execution time of clustering algorithms (for video data sets)

| **Clustering Algorithm** | **Clustering Methodology** | **Video 1 [3]** | **Video 2 [2]** |
|---|---|---|---|
| NMSC (proposed) | Non-Recursive | 0.02 ms | 0.02 ms |
| Online Small Group [5] | Recursive Agglomerative | 6.23 ms | 6.43 ms |
| ETH Flock Detection [6] | Recursive Spatio-Temporal | 0.81 ms | 0.82 ms |

complexity which was explained earlier - the NMSC is of $O(n^2)$ while the other two clustering algorithms is $O(n^3)$. Table 4.6 gives the clustering methodology adopted in the compared algorithms in addition to the mean execution times.

The NMSC algorithm has faster execution times than the other clustering algorithms. This is because it performs clustering in a single iteration (with a single similarity graph calculation) for each video frame. This approach is different from the recursive clustering algorithms adopted in [5, 6], where a termination criteria must be satisfied for clustering to stop. Recursive algorithms in [5, 6] identify clusters at each iteration based on the similarity values. Similarity values are calculated repeatedly to cluster potential clusters and individuals, to form updated clusters. Such an approach leads to numerous iterations of similarity calculations to satisfy the termination criteria at every video frame. Hence, the clustering algorithms in [5, 6] have longer execution times.

Table 4.7: Group match rate comparison

| | NMSC (proposed) | Online Small Group [5] | ETH Flock Detection [6] |
|---|---|---|---|
| a. Total number of unique groups | 118 | | |
| b. Identified groups | **103** | **78** | **59** |
| c. Missed groups | 15 | 40 | 59 |
| 1. Due to high occlusion | 11 | 17 | 18 |
| 2. Due to large distance between pedestrians | 4 | 23 | 41 |
| d. Group match rate (%) | **87** | **66** | **50** |

### 4.3.6 Group Identification Match Rate using NUSME Data set

The NUSME data set (explained in sub section 4.3.1) is used to compare the group identification match rate of the proposed NMSC algorithm and the existing clustering algorithms [5, 6]. Table 4.7 lists the group identification match rate of the clustering algorithms against the interview ground truth. The group match rate is calculated by comparing the number of identified groups against the total number of unique groups seen in the video. It is observed that the proposed NMSC algorithm has a match rate of more than 87 percent while the other algorithms have a group match rate of lesser than 70 percent. As explained in Sec 4.3.3, group results for [6] have a low match rate because of the deficient clustering threshold based on euclidean distance. Clustering algorithm in [6] does not consider the direction of travel for clustering and identifies pedestrians as groups even when they travel in different directions but close to each other. The clustering algorithm in [5] cannot identify sparse groups because of the strict cluster termination criteria. The algorithm cannot identify groups of larger size (i.e. group size greater than 3) and bigger intra-group distance. The termination criteria ensures clustering of a pedestrian to a group only when the pedestrian is related (similar in motion) to at least half of the other pedestrians in the group. Pedestrians who do not satisfy this criteria remain as individuals. Many instances of sparse groups are observed in the NUSME data set. An instance of group identification is displayed in Figure 4.8.

Figure 4.8: A group identified by the NMSC algorithm.

## 4.4 Summary

A novel non-recursive pedestrian clustering algorithm is proposed. The proposed NMSC algorithm is able to identify social groups in real-time. The motion similarity based clustering algorithm clusters pedestrians in a single iteration per video frame and does not adopt a recursive approach like the existing works. Adoption of the triangle inequality property in clustering results in an unbiased clustering, where connectivity between all group members is not a necessary condition. The group identification accuracy of the NMSC algorithm is compared to that of the existing similar algorithms using standard research data sets and also using synthetic data sets. Experimental results show that the proposed clustering algorithm performs on par or better than the existing techniques in terms of real-time performance (even at high crowd densities) and group identification accuracy.

# Chapter 5

# Pedestrian Group Feature Extraction

Law enforcement agencies rely heavily on visual surveillance to maintain law and order. Governments and security companies deploy and maintain massive infrastructures of Closed Circuit Television (CCTV), smart and Internet Protocol (IP) cameras for this purpose [93]. Such infrastructures generate enormous amount of data, reaching up to several terabytes of data every hour. Storing this enormous amount of data and retrieving useful information is highly expensive and painstakingly difficult. Hence, there is a growing need for intelligent, analytical methods to identify events and store only the corresponding video clippings and simple visual representations to understand the spatial and temporal information in those video clippings. From a security point of view, there are numerous events which might be of interest such as pedestrian meetings, pedestrian groups splitting, pedestrian running, lying down and many more.

In this thesis, pedestrians meeting[1] and split[2] events are considered to be the most important events. Pedestrians meeting and split events can be discovered by identifying and tracking pedestrian groups. The first section in this chapter introduces a data structure to record and store information related to pedestrian groups. This information is used in a Real-time Pedestrian Meetings and Visits Identification System (RPMVIS) to identify pedestrian meeting and split events. The second section explains several visualizations proposed to summarize the spatial and temporal information related to pedestrian groups.

[1]When pedestrians or groups of pedestrians merge, a pedestrian meeting event happens.
[2]When a pedestrian group splits, a pedestrian split event happens.

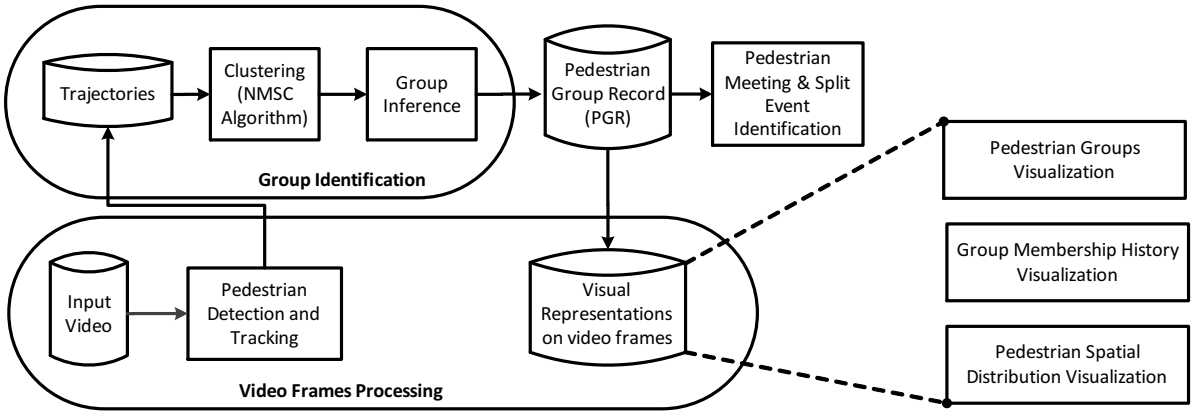



Figure 5.1: Block Diagram of Real-time Pedestrian Meetings and Visits Identification System (RPMVIS).

These visualizations help to identify the pedestrian groups in real-time, identify history of pedestrian meeting events and discover frequently visited areas in the video frame region. The RPMVIS is explained in the next section.

## 5.1 Real-time Pedestrian Meetings and Visits Identification System (RPMVIS)

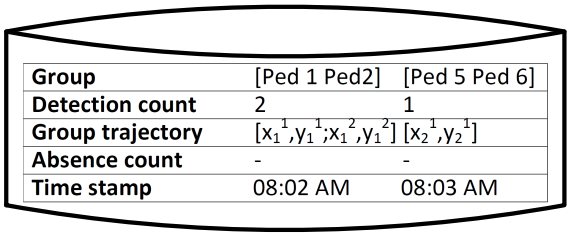
The proposed real-time system automatically identifies pedestrian meetings and visits from surveillance videos. The system has a Pedestrian Detection and Tracking module, Pedestrian Group Identification module (NMSC algorithm in sub-section 4.2.2), a Pedestrian Group Record (PGR) and three visualizations of the PGR information. The block diagram of the proposed real-time system is shown in Figure 5.1. Pedestrians are detected and tracked by the pedestrian detection and tracking module (Sec. 3.1.1). The pedestrian trajectories are pairwise compared, using the pedestrian group identification module, to identify pedestrian groups. Identified groups are logged in the PGR with several features (start time, location of meeting, picture of meeting, group detection count, group absence count). Pedestrian meeting events are automatically identified from the PGR. The PGR is utilized to generate three visualizations: the pedestrian groups, group membership history (with video indexing) and the spatial distribution of the pedestrians.

The real-time system has been deployed in residential halls at the National University of Singapore (NUS). The camera-server topology for the RPMVIS is also explained. The real-time challenges such as the camera-server communication lag, faced during deployment of the system are discussed. Since the pedestrian detection-tracking module and the pedestrian group identification modules have been discussed in the earlier chapters, only the pedestrian group record and the steps to identify the pedestrian meeting and split events are explained in the following subsection.

### 5.1.1 Pedestrian Group Record (PGR) and Pedestrian Meeting and Split Event Identification using PGR

There are several features which are extracted from the process of pedestrian group identification, namely group size, group member identity, group members' trajectory, detection count, absence count and time stamp at which the group is identified (Figure 5.2). Group size is the number of pedestrians in the identified group. Group member identity is a unique string (such as Ped1, Ped2) assigned to every member in a group. Group members' trajectory is the sequence of centroid coordinates of the group members. Detection count is the number of video frames in which a group is identified. The detection count is incremented by a value of 1 each time (i.e. every video frame) the group is identified. The detection count can be used to determine the time duration for which the group was present in the video frame region. Pedestrian groups may not be identified at every video frame. Issues like high occlusion, pedestrians momentarily moving away from each other lead to pedestrian groups momentarily not identified. Such situations may not signify an actual splitting of pedestrian groups. In order to identify the actual pedestrian group splitting events, an absence count[3] is recorded for every identified pedestrian group. The absence count is incremented by a value of 1 only when a group is not identified. The group which has an absence count of less than 50 frame (i.e. 2 seconds in a 25 frames/second video), remain active in the PGR. If the absence count exceeds 50, the corresponding group is considered to be split and the future group identifications are

[3]Absence count is the number of video frames over which the group (which was previously identified) is not identified.



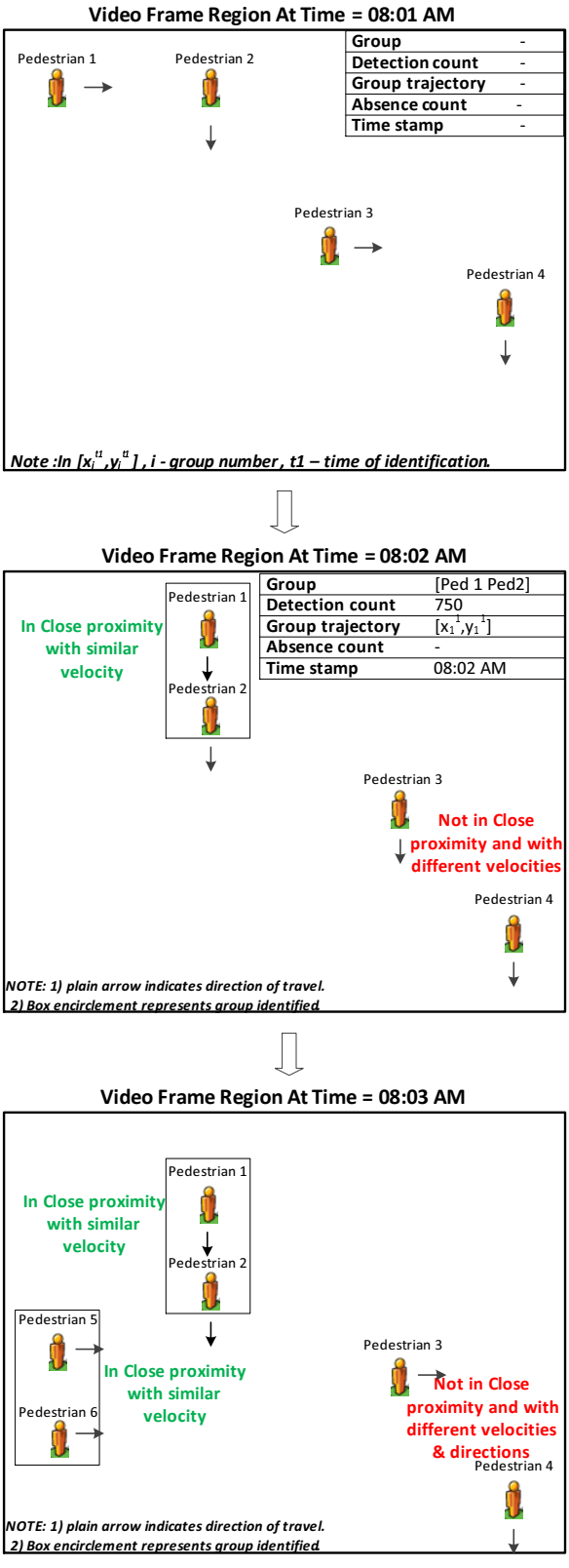
| Group | [Ped 1 Ped2] | [Ped 5 Ped 6] |
|---|---|---|
| Detection count | 2 | 1 |
| Group trajectory | $[x_1^1, y_1^1; x_1^2, y_1^2]$ | $[x_2^1, y_2^1]$ |
| Absence count | - | - |
| Time stamp | 08:02 AM | 08:03 AM |

Figure 5.2: Structure of the data in the Pedestrian Group Record (PGR).

treated as a new group with a new entry created in the group record. These features describe the identified pedestrian groups across time. These spatial and temporal features of pedestrian groups are extracted and stored in the proposed Pedestrian Group Record (PGR). The features stored in the PGR form the basis for several applications explained in the subsequent chapters.

A typical PGR is explained with the help of Figure 5.3. In this example, pedestrians 1 and 2 form a group at 8.02 am and pedestrians 5 and 6 enter as a group at 8.03 am, walking closely with similar velocity. Pedestrians 3 and 4 do not form a group as they are not spatially close and walk with different velocities and directions. When a group is identified, a new group entry is created. The first field in the entry carries the group member identities (for example, e.g. $[Ped1\, Ped2]$). The size of this field represents the size of the group identified (group size is 2 in the example). The detection count field is incremented by 1 each time (i.e. every video frame) the group is identified. The group members' trajectory field is updated with the 2D coordinates of the group members. The absence count field is updated only when a group is not identified.

Pedestrian events related to pedestrian groups can be identified using the PGR. When pedestrians or groups of pedestrians merge, a meeting event happens. When a pedestrian group splits, a split event happens. The flow chart in Figure 5.4 explains the steps to identify a meeting or a split event using the PGR. A meeting event is identified when a newly formed group's existence is confirmed. A group's existence is confirmed when the new group's detection count exceeds a pre-defined group persistence threshold. A split event is identified when a group's termination is confirmed. The group's termination is confirmed when the group's absence count exceeds a pre-defined group absence threshold. The threshold values are empirically identified for the video scenes. The group persistence




| Group | [Ped 1 Ped2] | [Ped 5 Ped 6] |
|---|---|---|
| Detection count | 2250 | 500 |
| Group trajectory | [$x_1^1$,$y_1^1$;$x_1^2$,$y_1^2$] | [$x_2^1$,$y_2^1$] |
| Absence count | - | - |
| Time stamp | 08:02 AM | 08:03 AM |

Figure 5.3: Group recording in PGR with time.

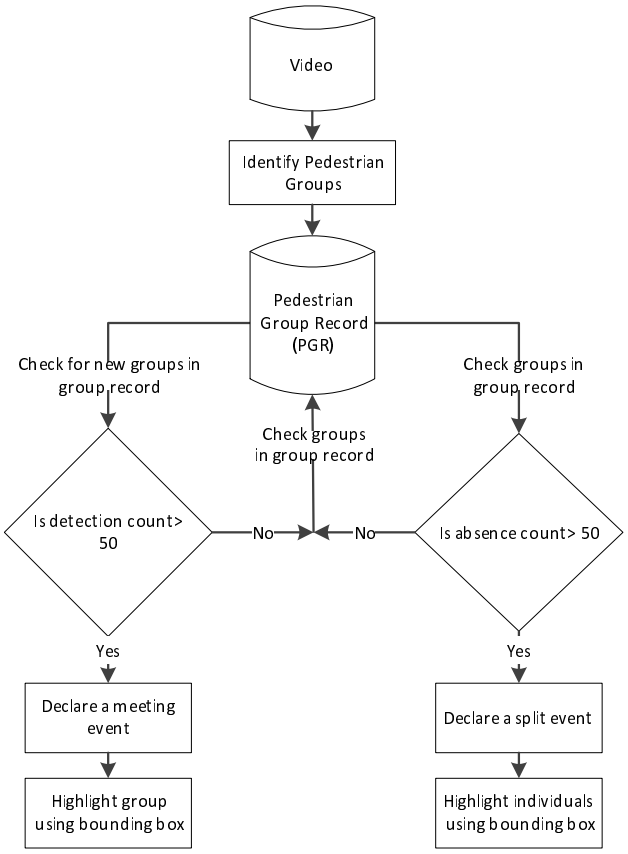



Figure 5.4: Flow chart illustrating the steps for pedestrian Meeting and split event identification using the PGR.

threshold and the group absence threshold are set as two seconds (50 frames for a 25 frames/second video). For this video, the group persistence threshold was set at four seconds and group absence threshold at 5 seconds. A meeting event identified in the NUSME video data set is highlighted in Figure 5.5. The performance evaluation of the RPMVIS is explained in the following section.

## 5.2 RPMVIS Event Identification Evaluation

Several aspects of the NMSC algorithm (the pedestrian group identification module in the RPMVIS) are evaluated, namely: performance against human observer's judgment (using Kappa score measurement), computational complexity and real-time performance

Figure 5.5: A meeting event identified by the RPMVIS, highlighted by a blue box.

when compared to that of existing pedestrian grouping algorithms [5, 6]. The evaluation results for the algorithm are discussed in Sec. 4.3. The event identification accuracy is discussed in this section.

Ten-day video feeds form four camera locations (Figure 5.6) in the NUS are manually observed to build the ground truth of pedestrian events. The RPMVIS identifies pedestrian events, namely meeting and split events (explained in sub-section 5.1.1). The event identification match rate is calculated by comparing the identified events against the ground truth of the meeting and split events. A human observer was employed to build the ground truth of events. Table 5.1 shows that the RPMVIS has a high pedestrian group identification accuracy. The camera locations are at common areas at residential halls in NUS. The locations are monitored for student's activities such as unauthorized gathering in the common areas, dancing and any activities during midnight. Events which happened on rainy nights (at camera locations A, B and C) could not be identified due to the pedestrians not being detected continuously. Events which involved large number of people such as meeting during dancing sessions could not be identified because of high pedestrian occlusions.

### 5.2.1 Camera-Server Topology

Figure 5.7 outlines the camera-server topology adopted. A typical video surveillance infrastructure consists of numerous security cameras, command centers and servers (for video archival). All these components are connected together through a local area network using ethernet cables. The existing infrastructure of the cameras and the control center


Figure 5.6: Camera locations selected for testing RPMVIS.

Table 5.1: Event identification accuracy across a 10-day video feed for four camera locations, events refer to the meeting and split events of pedestrians

| Event Information | Camera Locations | | | |
|---|---|---|---|---|
| | A | B | C | D |
| Total Number of Unique Events | 52 | 24 | 18 | 31 |
| Identified Events | 48 | 22 | 15 | 28 |
| Missed Events | 4 | 2 | 3 | 3 |
| Event Match Rate (%) | 92 | 91 | 83 | 90 |

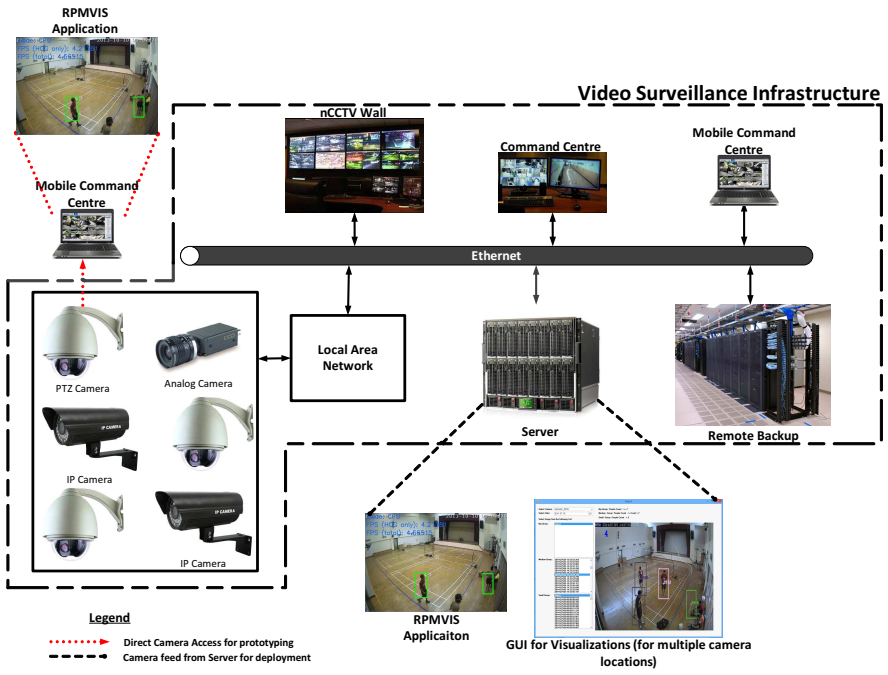



Figure 5.7: The Camera-Server Topology. The RPMVIS was housed in a laptop during prototyping and then in a server for deployment.

with the servers are used in this application without any modification. In this pilot study, the RPMVIS was housed in a laptop directly connected to the security cameras (indicated by the red arrows in Figure 5.7). The server was setup with the system for real-time deployment (indicated by the short dashed lines in Figure 5.7). The Software Development Kit (SDK) of the CCTV system (Verint) captures the video frames from security cameras to create video for archival. Every video feed is processed by it's own instance of the RPMVIS, creating and populating a separate Pedestrian Group Record for that video feed.

A GTX 670 NVidia graphic card was used for the Fast HOG (for pedestrian detection, using the Graphics Processing Unit (GPU)) implementation to process a maximum of 18 video feeds simultaneously. Each video feed is processed at 45 Frames Per Second (FPS) on an average. The same number of video feeds are processed at less than 4 FPS with a normal HOG implementation when using only a processor (Intel I7). The RPMVIS is scalable and can handle up to 18 video feeds, simultaneously with the use of GPUs in

the graphic card. The processing speed reduces below the normal video frame rate of 25 FPS, when more than 18 video feeds are processed by the system. The reduction in the processing speed is because of the hardware's processing limitation. More number of video feeds can be processed with a better hardware configuration. The calibration performed in the RPMVIS to overcome real-time performance challenges when camera feeds are accessed through internet are explained in the next section.

### 5.2.2 Real-time Performance Challenges

There are many real-time challenges while identifying pedestrian meeting events with the RPMVIS. The two main critical challenges are loss of video frames in the transmission medium and intermittent communication failure. These challenges are observed to occur only when the camera feeds are accessed through the internet. This section explains the reasons for these problems and the calibration performed to address them.

**Calibration of NMSC algorithm for Low Video frame rates** Pedestrian groups are identified in real-time to detect pedestrian meeting events. During the pilot study the cameras were directly connected to the RPMVIS. Such a direct connection results in no video frame loss. During deployment in the residential halls at NUS, the cameras were not allowed to be accessed directly due to security reasons. The SDK of the video surveillance system could not be utilized to access the video frames in real-time and hence a separate application was built to capture the video frames in real-time. The application utilizes the existing transmission medium to capture the video frames. The transmission medium includes the server from which the video frames are sent to the RPMVIS over the internet. Many video frames are lost in the transmission medium from the camera to the RPMVIS[4]. Low internet bandwidth[5] and high internet traffic are some of the factors

[4]The RPMVIS was tested with the offline video feeds and video streams directly accessed from the IP cameras. This test was performed to check whether the system misses any video frames from processing. It was identified that all the video frames were processed without losing any of them. When the video feeds are accessed through the transmission medium, it was observed that many video frames were lost. The video feed at the receiver side has empty video frames when the actual video frames are lost.

[5]The average internet bandwidth of the NUS transmission medium is 3270 Mbps for download operation and 907 Mbps for upload operation [94]. This bandwidth is a shared resource for the students, staff in the university and several video surveillance systems for video archival IP camera feeds. Hence, the

which lead to loss of video frames [95]. The application utilizes the existing transmission medium because it will be a huge cost to build a separate transmission medium. The shared transmission medium (with a low internet bandwidth) is suspected to be the main reason for video frame loss.

Based on an empirical study of the application, it was identified that at most three video frames per second could be captured consistently. The effect of such low frame rates on pedestrian group identification accuracy was analyzed. Figure 5.8 highlights the NMSC algorithm's group identification accuracy at low video frame rates. In Figure 5.8, a false negative refers to a pedestrian group not identified during its period of existence, under sampling refers to the low video frame rates. The NUSME data set (Sec. 4.3.1) is used for this analysis. Figure 5.8 outlines the performance degradation of the NMSC algorithm with decreasing number of video frames available per second. From Figure 5.8, it was concluded that pedestrian group identification accuracy at 3 frames per second is not drastically reduced compared to the accuracy with 25 frames per second. The NMSC algorithm in the RPMVIS has to be calibrated to work at this low video frame rate. The calibration involves changing the tracklet size for pedestrian clustering. The reason to change the tracklet size is explained as follows. The tracklet size in the NMSC algorithm is chosen as 30 video frames for a 25 frames per second. This setting is performed to consider pedestrian centroids from only the past one second for pedestrian clustering. When the frame rate is 3 frames per second, the same tracklet size cannot be used any more because the tracklet with 30 video frames will hold centroid information of past five seconds. Hence, the tracklet size is reduced to 2 video frames to consider pedestrian centroids from only the past one second for pedestrian clustering.

**Testing RPMVIS System for intermittent communication failure** The RPMVIS's response to intermittent communication failure is tested. The communication failures lead to temporary loss of video frames. Individual IP cameras are connected to a computer housing the RPMVIS and the following communication failure scenarios are conducted.

bandwidth usable per user is limited in the NUS transmission medium.

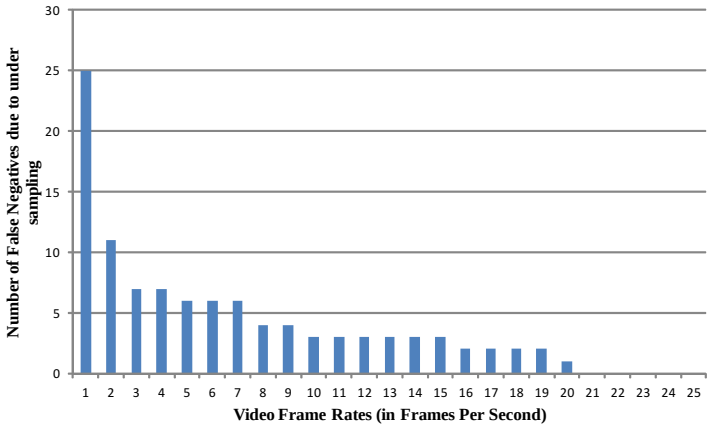



Figure 5.8: Pedestrian groups missed (false negatives) by NMSC algorithm at different video frame rates. Here, a false negative refers to a pedestrian group not identified during its period of existence, under sampling refers to the low video frame rates.

- Intermittent LAN port failure by temporarily disconnecting the LAN connection at the IP camera side and / or computer's side.

- Intermittent power failure at the IP camera side by temporarily disconnecting the power cable.

The above cases resulted in loss of video frames from the IP camera during the period of intermittent failure, but the system was able to proceed with video frame processing after the failure. Different periods of temporary disconnection were tested, from a few milliseconds to 5 seconds. The system was observed to continue (more than 90% of the times) with video frame processing after intermittent communication failures whose time duration is lesser than three seconds. This behavior of the system is explained as follows. The application to capture the video frames utilizes a video codec shell script which polls for new video frames. The script has a fixed threshold time period beyond which polling is terminated, if new video frames are not captured. The threshold period is approximately three seconds. Hence, the system crashes due to communication failures which last for longer than three seconds.

Figure 5.9: Pedestrian Groups Visualization - A visualization on the video frames. Pedestrian group membership highlighted by colored bars (over pedestrian head).

## 5.3 Visualizations

Visualizations of spatial and temporal pedestrian information are useful when searching for events in video surveillance. The pedestrian groups visualization helps to monitor the pedestrian events in real-time. The group membership history visualization helps to identify the history of pedestrian events that happened. Security personnel no longer need to watch the entire video, if the pedestrian group records are stored offline and can be retrieved later for these visualizations. The visualizations are explained below.

### 5.3.1 Pedestrian Group Visualization

The pedestrian groups visualization highlights the pedestrian groups in the video frame in real-time. When the pedestrian groups are identified, group members are highlighted on the video frame with the representation shown in Figure 5.9. Each person is assigned a colored bar, displayed near his / her head. The color of the bar indicates the group to which they belong and the width of the bar provides a relative measure of the period of the group membership.

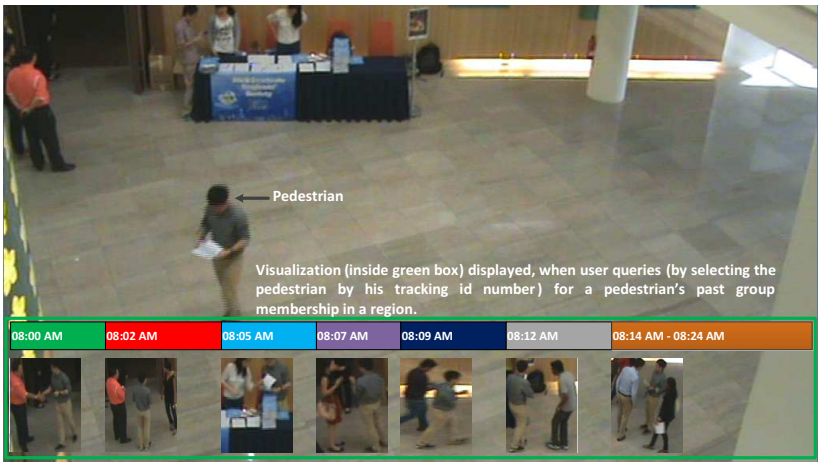



Figure 5.10: Group Membership History Visualization (with Video Indexing) - A visualization of individual pedestrian's meeting history. The visualization is within the green box.

### 5.3.2 Group Membership History Visualization

The group membership history visualization highlights individual pedestrian's group membership history. In order to understand and visualize pedestrians' group history over a period of time, it is necessary to have a visualization which displays the past meeting information. The pedestrian's group membership history visualization shown in Figure 5.10 collates the past group information into an easily understandable format. The visualization is highlighted by a green box in Figure 5.10. In the visualization, the color of the bars discriminate among groups, the width of the bars provides a relative measure of the period of group membership. The meeting start time (video indexing) is part of the bar along with a picture of the group members. When a user queries for a pedestrian's past group membership in a region, the user selects a pedestrian by his or her group member identity from a video frame and the corresponding visualization is displayed. The user no longer needs to observe the entire video to find that pedestrian's meetings.

These visualizations can be applied for multiple regions (with multiple video feeds covering adjacent locations) by tracking pedestrians across the different regions (using methods like facial recognition [57] and face tracking [58]).


Figure 5.11: 'E-Resources Discovery Day' event regions of interest on day 1 (left) and day 2 (right) .

## 5.4 Determining Stall Occupancy using Pedestrian Spatial Distribution Visualization

The spatial distribution visualization can be used to monitor regions where large number of groups of pedestrians are present. For example, in this application, it was used to identify frequently visited regions which were under surveillance. The video recording of 'E-Resources Discovery Day' event held in National University of Singapore was monitored to demonstrate the visualization. The event has several attractions like E-Resources stalls, book sales, garage sales and fun games for the public. It was conducted over two days with thousands of people attending the event. Two camera viewpoints (Figure 5.11) were used to record two different regions. Over six hours of the event was video recorded over the two days. The crowd density was high with more than 20 pedestrians within the regions at any time.

In this video, the popularity of shopping outlets (stalls) was understood by visualizing the pedestrian visits over a period of time. Pedestrian group members' pixel locations are queried from the PGR and segregated based on the group sizes. This query process is repeated for every video frame for over a period of time to build an ensemble matrix which accumulates the number of group member detections at each pixel location. The matrix has a dimension equal to the video resolution (for example, the ensemble matrix size for a 112×160 video is 112 rows by 160 columns). The ensemble matrix is utilized to generate a contour plot with a heat map representation. The heat map representation

**Algorithm 5.1** Ensemble Matrix $E_s$

```
for (i = 1; i <= last video frame; i++)   "For every video frame (1 to last video frame in video)"
|   for(j = 1; j <= l; j++)              "For all groups (1 to l) identified in each video frame"
|   |   s = size(G^j);                  "Determine the size (number of group members) of the group G^j"
|   |   for(k = 1; k <= m; k++)         "For all the group members (1 to m)"
|   |   |   [x, y]= centroid(G^j[k]);   "Determine the centroid of the group member G^j[k]"
|   |   |   E_s[x, y]= E_s[x, y] + 10;  "Populate the corresponding ensemble matrix's element"
|   |   end
|   end
end
```

assigns light colors (colors close to white) for pixel locations which have more number of pedestrian group member detections. When the contour plot is superimposed over the video frame region, the stall occupancies with respect to the group sizes are discovered. These superimposed plots are termed as the pedestrian spatial distribution visualizations. The RPMVIS identifies individual as well as pedestrian groups of sizes 2, 3, 4 and 5. Hence, separate ensemble matrices for each group sizes are built as discussed earlier.

The ensemble matrix is represented as $E_s$, where $s = 1, 2, 3...5$, representing the group sizes. Algorithm 5.1 is adopted to populate ensemble matrices using information from the PGR. The PGR is queried for pedestrian groups identified at every video frame. The size of the pedestrian groups are determined from the PGR. The centroid of a group member corresponds to a particular element in the corresponding ensemble matrix. This particular element's value is incremented by ten. This process is repeated for all group members.

Figure 5.12 shows the visualizations generated for different group sizes (group size 5 at the top to individuals at the bottom) for the two regions. The visualization represents the stall area occupancy with lighter colors (in the heat map) representing higher area occupancy. Based on the visualization for the first day, the E-Resources stall had lesser attention than the cashier desk, Freebie / information counter and garage sales. People visited the event more as individuals and in groups of size less than 4. Groups of size 4 and 5 were least detected. For the second day, the Chinese book stall was most visited in the observed period with considerable attention to books on hobby / leisure. People visited this section of the event more as individuals and in groups of size less than 4.

Figure 5.12: Stall occupancies during 'E-Resources Discovery Day' event, day 1 (left), day 2 (right). Group size 5 at the top to group size 1 (individuals) at the bottom.

Groups of size 4 and 5 were least detected.

These observations do not provide information on the purchasing interests of the people, but provides information about the popular stalls of the event and how people (in groups or as individuals) visited the stalls. This information coupled with human intelligence in reasoning these occurrences (with external factors not available to the video analysis) might be helpful in future planning. A possible change to increase visit to the E-Resources stalls could be to move all the E - Resources stalls forward before the other sections.

## 5.5 Summary

A Real-time Pedestrian Meetings and Visits Identification System (RPMVIS) using the Non-recursive Motion Similarity based Clustering (NMSC) algorithm is proposed. Using the proposed Pedestrian Group Record (PGR), the RPMVIS is able to identify pedestrian meeting events in real-time with high accuracy. The real-time system's scalability is demonstrated by the deployment in a residential hall within National University of Singapore. The proposed visualizations help to understand pedestrian visits and discover popular areas within regions. From these works, the pedestrian group features in the PGR prove to be useful to understand the pedestrian activities.

Identifying pedestrians in groups by employing facial recognition will help to continue tracking pedestrians who move from one region to another. The relevance of this system and its information about pedestrian meetings can be explored in different scenarios, which can provide insights on how pedestrians behave in different environments. In scenarios such as surveillance applications, the automatic detection of build-up of unusually large groups of people can alert security personnel of impending issues. A commuter friendly application in a taxi queue scenario can be identification of pedestrian groupings which take place as passengers arrive at the taxi stand. This information can be used to suggest a taxi type: big, medium and small taxi. Such commuter centric applications are explored and analyzed in the next chapter.

**Acknowledgment** The author would like to thank the NUS Sheares Hall, NUS Libraries for providing access to the video footage for the experiments and the NUS Ambient Intelligence (AMI) lab in Interactive Digital Media Institute (IDMI) for their support in carrying out this research work.

# Chapter 6

# Applications Based on Pedestrian Group Identification

Pedestrian groups are identified by the pedestrian group identification technique explained in Chapter 4. Pedestrian group features such as group size, group member identities, group members' trajectories are extracted and stored in the Pedestrian Group Record (PGR). This chapter explains several applications which use these features to generate useful information for commuters who use public places and public modes of transport. Applications such as people counting at taxi queues, and food outlet queues are proposed to determine the crowd level in these queues. People counting at door entrances and exits is performed to determine the number of people entering and exiting a location. A prototype solution to determine crowd density at train platforms is proposed and tested with simulated videos. This solution serves as a proof of concept for utilizing group member features for crowd density estimation. The applications are discussed below.

## 6.1 Video Based People Counting

Video Based People Counting determines the number of people in a video scene. A simple framework is proposed for people counting in Figure 6.1. Based on the framework, a region of interest (ROI) is first marked in the video frame region. There are different ROIs depending on the application. The entire video frame region is the ROI if the application

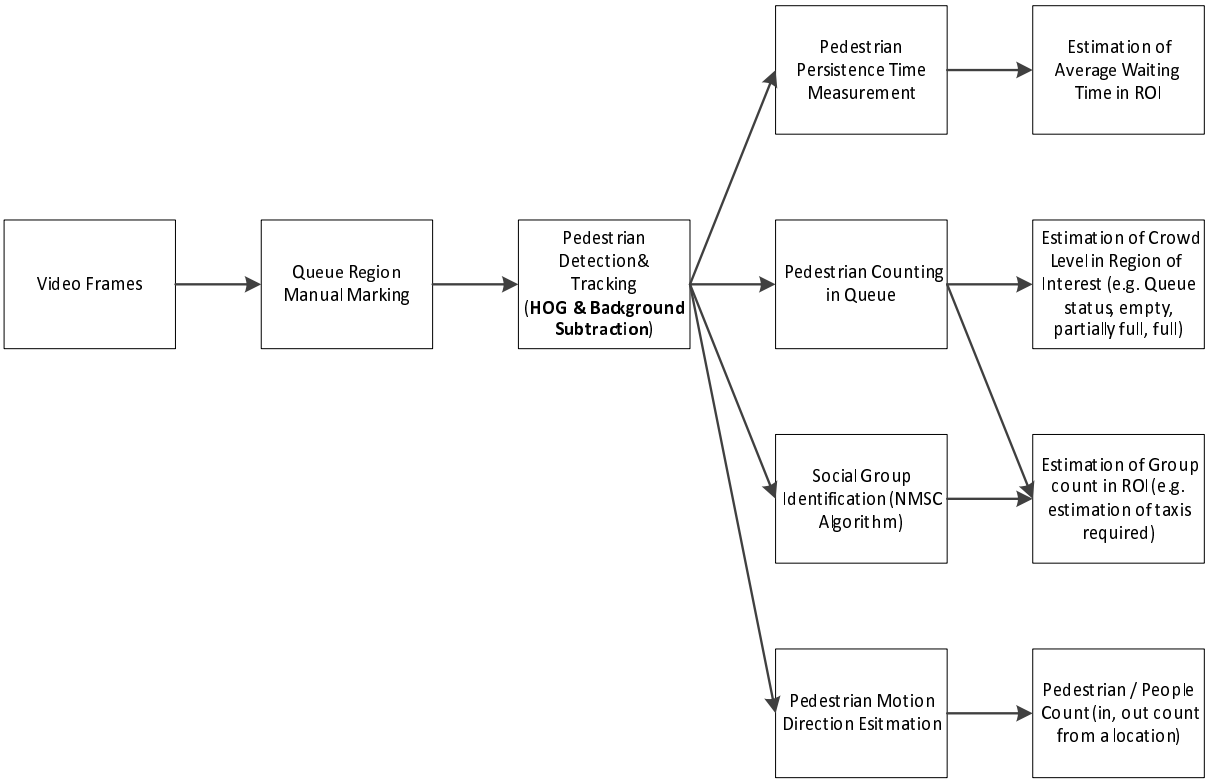



Figure 6.1: Proposed framework for video based people counting.

involves people counting at public places such as fairs. The queue region is the ROI for people counting at queues and door entrance area is the ROI for people counting at the door entrances. The ROI has to be identified because it is necessary to determine the people count only in this region and not consider other regions. Pedestrians are detected and tracked in the ROI to count the number of people in the ROI. The people count is compared against the maximum capacity of the ROI to determine the crowd level in the ROI. Three crowd levels are considered: empty, partially full and full. The measurement of pedestrian persistence time in the ROI is possible because of pedestrian tracking. An average of the pedestrian persistence time values represent an estimate of waiting time in the ROI. The crowd level and the waiting time are useful parameters in ROIs which involve queuing.

In the following subsections, video based people counting is applied to queues in taxi stands, food outlets and the library.

### 6.1.1 People Counting at Queues in Taxi Stands and Food Outlets

Preliminary tests of the people counting algorithm was performed on a taxi queuing region (Figure 6.2). Due to the unavailability of actual real-world videos, a simulated taxi queue environment is created using the PTV VisWalk software[1] [96]. The environment (Figure 6.2 (left)) is created with three pedestrian sources. The queuing area (Figure 6.2 (middle)) is designed to be 10×0.5m, with the maximum possible crowd density of up to 30 people in the queue. The crowd density is set to random to resemble the real-world environment, where the crowd flow is uncontrolled. Two of the pedestrian sources are configured to originate from the buildings in the environment. And the third pedestrian source is configured to originate from the pedestrian pavement. A random flow of taxi traffic is created and pedestrians who queue, board the taxis. So, when the pedestrian queuing increases or the taxi traffic reduces, the taxi queue becomes crowded. Two view points (displaying different areas in the environment) are selected from the taxi queue environment, namely, taxi queue entrance area (Figure 6.2 (right)) and the queuing area (Figure 6.2 (middle)). Social groups are identified from the queue entrance area video, using the NMSC algorithm explained in Sec. 4.2.2. The video recorded from the queuing area is used to count the number of pedestrians in the queuing area, thereby determining the crowd level in the queuing area.

The steps in the queue people counting algorithm are explained in the following box. The algorithm outputs the people count in the queuing region and the average waiting time.

[1]PTV Viswalk simulates and models the human walking behavior along with road traffic. Planners use this industrial software tool whenever pedestrian flows need to be simulated and analyzed for indoor and outdoor locations. The software provides realistic results, modeling any number of pedestrians in 2D and 3D environments. It is built with various pedestrian behavior models like queuing behavior, panic behavior, aggressive behavior and so on. The software has a video recording option, which is used to record videos which are used in place of the actual videos.

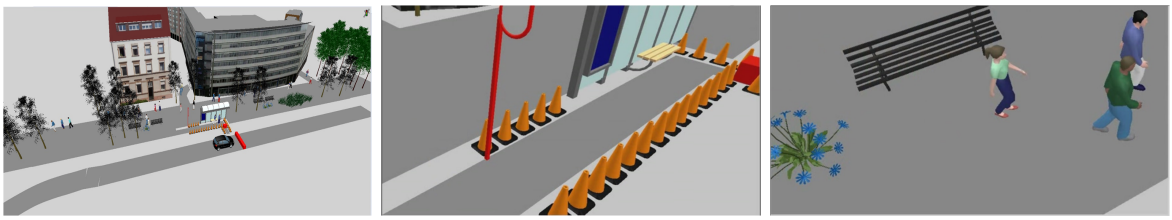


Figure 6.2: Taxi queue environment (left), taxi queuing area (middle), taxi queue entrance area (right).

| Steps in queue people counting and waiting time estimation |
|---|
| 1) Mark the queue region |
| 2) For every video frame |
| 2.1) Detect and track pedestrians within the queue region |
| 2.2) Count the number of pedestrians in the queuing region |
| 3) Record entry and exit time of pedestrians to measure the persistence time |
| 4) Calculate average waiting time using the persistence time measured in the recent past (e.g. last 10 minutes) |

Figure 6.3 outlines the high level of accuracy in pedestrian counting. The results are shown for five simulation videos recorded with random crowd density. Each video has a time duration of 50 minutes. The ground truth of number of pedestrians was manually recorded by a human observer. Figure 6.4 shows the output of the people counting displayed as a queue status, with queue status levels ranging from empty to full. The information on the number of taxis required is based on the group identification results (using the NMSC algorithm) applied to the taxi queue entrance area (Figure 6.2).

The first video clip (Video 1) was utilized to validate the estimated pedestrian waiting time at the taxi queue against the actual waiting time. Figure 6.5 shows the consistency of the average time estimates against the actual waiting time (ground truth). The estimated waiting time deviates (at few time instances) from the actual waiting time at very high crowd densities. It is observed that this deviation is a result of very high occlusion in the queuing region. The human observer manually recorded the actual waiting time of the

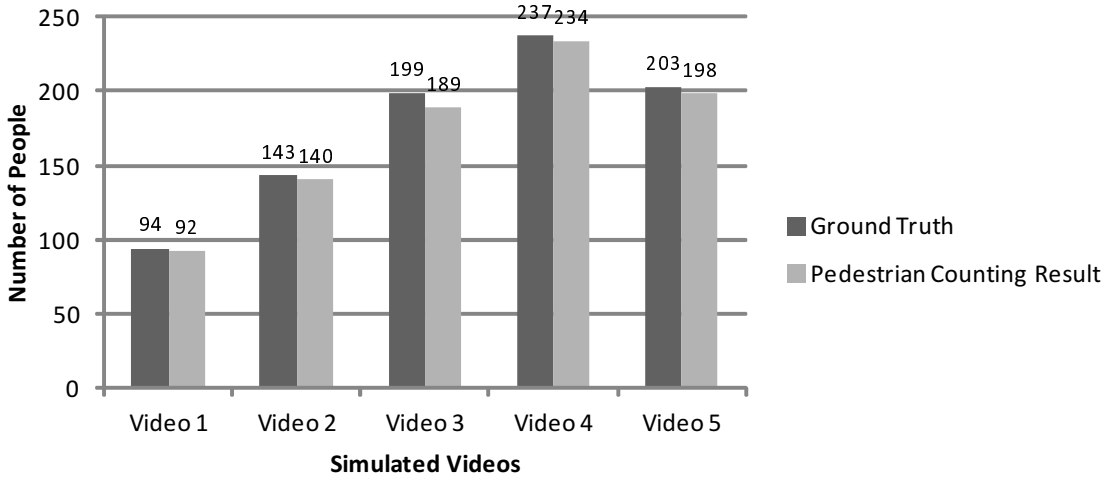



Figure 6.3: People counting results for taxi queue.

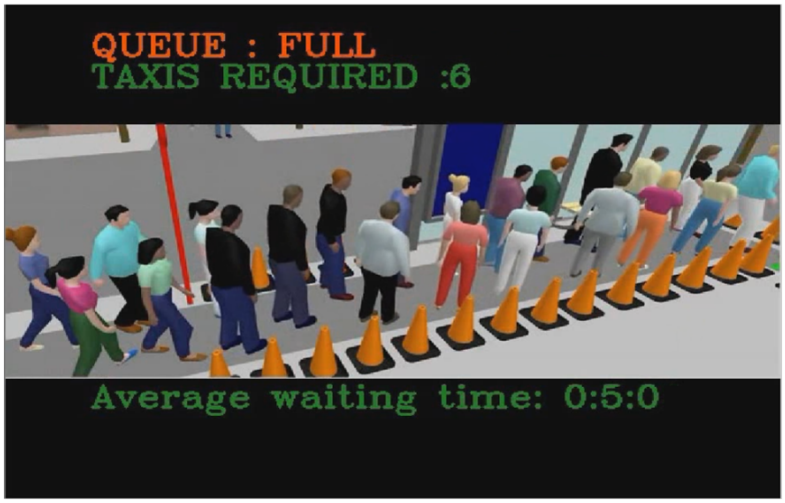



Figure 6.4: Crowd level in the taxi queue based on pedestrian counting, crowd level is displayed on the video stream.

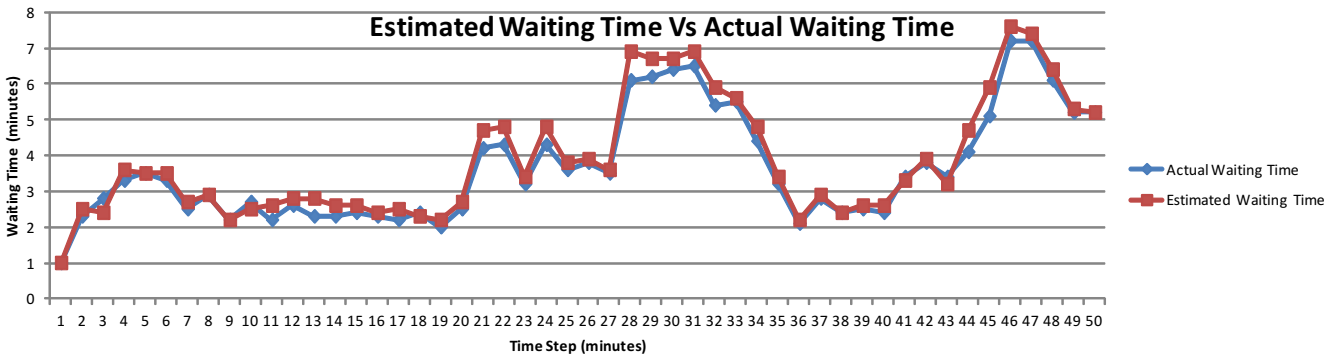



Figure 6.5: Estimated waiting time in a taxi queue across Video 1.

pedestrians using an electronic stop watch. The video was repeatedly observed to record the waiting time of all the pedestrians.

**Queue Region Marking** In most of the real-world video feeds, the camera has no means to capture the queuing region separately. Hence, the queuing region has to be marked in the video frame region to count the people who are queuing. In this thesis, two methods are explained to mark the queuing region. They are 1) manual marking of queue regions and 2) automatic marking by learning the spatial distribution of the queue members. The first method is manual and straightforward, where a human user marks the queue region based on experience gained by monitoring the video frame region. Figure 6.6 shows the manual queue region markings on the video frame. The second method is automatic, it is explained in the next paragraph.

People counting by manual marking of queue region may not always be an effective approach. Queuing regions have to be manually marked every time the people counting application is applied to a new location and manual queue region marking has to be repeated if the spatial structure of the queue region changes. So, a pragmatic method is proposed and tested to address this problem. This method automatically identifies the queuing region and people counting is performed within this identified region. The queuing region is automatically identified by learning the spatial distribution of the pedestrian group members. Queuing is a social behavior wherein people stand in close proximity to others to achieve a common objective. Hence, they form social groups. In case of


Figure 6.6: Queue regions in different food outlets with manually marked queue regions.

a food outlet, pedestrians form social groups in front of the food outlet. Such social groups in queuing regions are observed to exist for a longer period of time (minimum of 4 seconds) compared to momentary groups which occur elsewhere in the video frame region. Using the social group identification technique (Sec. 4.2), these groups are identified and their features (like group size, group member locations) are extracted and stored in the Pedestrian Group Record (PGR).

The stages in automatic queue region marking method are outlined in Figure 6.7. Group member locations are queried from the PGR to generate spatial distributions of the pedestrian group members over a period of time. Such distributions are learned by selecting the sample points from the spatial distributions which are detected in majority of the spatial distributions to build the spatial model. In this method, a sample point is selected for the spatial model, if it is detected in 4 out of the five spatial distributions. The spatial model is utilized to generate the queue region using Delaunay Triangulation [97] of the sample points in the spatial model. The performance comparison between the manual marking method and the automatic marking method is carried out using the different food outlet locations (Figure 6.6).

Figure 6.8 highlights the group members' spatial distributions generated utilizing the

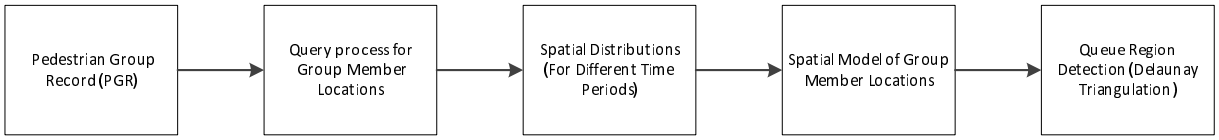



Figure 6.7: Stages in automatic queue region detection.

PGR information for the TC Noodles food outlet. Figure 6.9 (right) shows the queuing region automatically identified for the TC Noodles food outlet. Figure 6.10 outlines the people counting results for the two methods for five different time periods (each video clip has a time duration of 2 hours) in the queue location. The spatial distributions generated for the other three food outlets are shared in appendix B. It is evident that the automatic queue region detection performs on par to the manual queue region marking method.

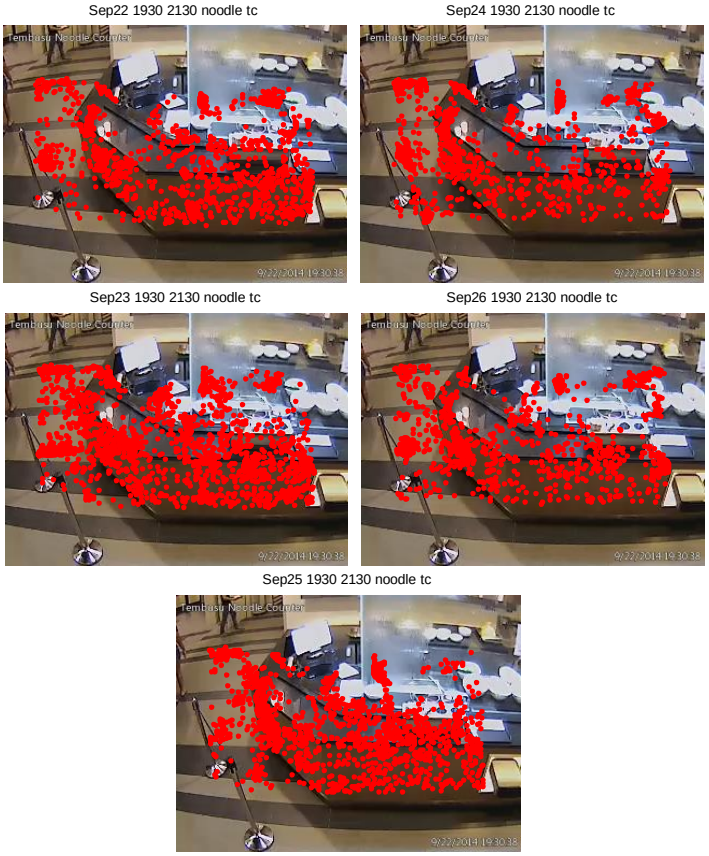



Figure 6.8: Group members' spatial distributions at five different time periods extracted from the Pedestrian Group Record (PGR) for TC Noodles Queue.

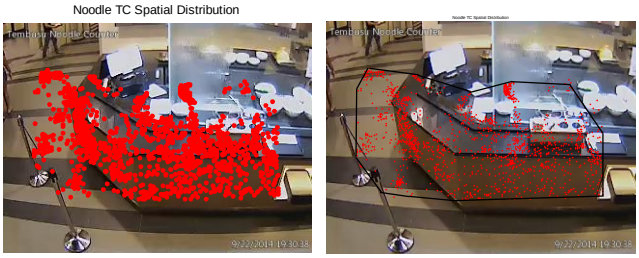



Figure 6.9: Spatial model (left) and identified queue region (right - marked by a black polygon) for TC Noodles Queue.

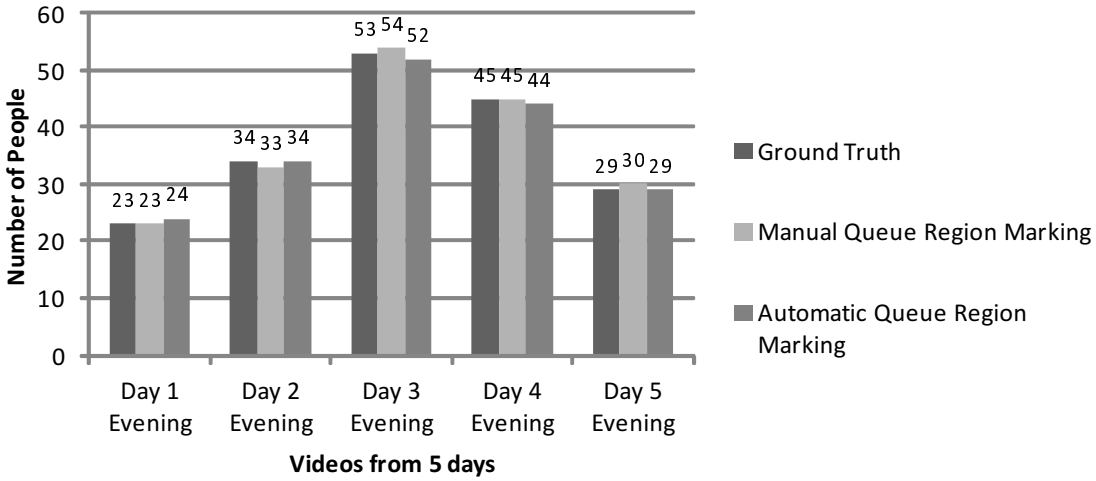



Figure 6.10: People counting results for TC Noodles queue.

### 6.1.2 People Counting at Door Entrance of Library

People counting at door entrances can be considered as an extension of the problem of people counting in queues. The number of people entering (in count) and the number of people exiting (out count) through a door entrance is determined. As explained in the people counting framework (Figure 6.1), the pedestrian's motion direction is estimated to determine whether a pedestrian is entering (in count incremented) or leaving (out count incremented) a location. The direction is estimated with respect to the origin point (which is the top left corner pixel, shown in Figure 6.11) of the video frame. The

**Algorithm 6.1** Direction estimation algorithm for people counting at door entrance

*For pedestrian entering Region Of Interest (ROI)*
| *Track pedestrian within ROI*
| $x_{diff} = x_{end} - x_{first}$; "Difference of the last and first x coordinates"
| $y_{diff} = y_{end} - y_{first}$; "Difference of the last and first y coordinates"
| $if(((x_{diff} < 0$ && $y_{diff} < 0)$ || $(x_{diff} > 0$ && $y_{diff} < 0)$ || $(x_{diff} == 0$ && $y_{diff} < 0)))$
| | *out direction*; "Moving towards origin point"
| $else\ if(((x_{diff} < 0$ && $y_{diff} > 0)$ || $(x_{diff} > 0$ && $y_{diff} > 0)$ || $(x_{diff} == 0$ && $y_{diff} > 0)))$
| | *in direction*; "Moving away from origin point"
| *else*
| | *do nothing*; "Person within the door region"
| *end*
*end*

pedestrian coordinate values are pixel distance values from the origin point. When a pedestrian's coordinates at two time instances are available, the sign of the difference between these coordinates will give the pedestrian's direction of motion with respect to the origin point (whether the pedestrian is moving towards the origin point or not). For the door location considered in Figure 6.11, moving towards the origin point represents moving out of the location and moving away from the origin point represents moving in to the location.

The pedestrian direction estimation algorithm (algorithm 6.1) utilizes the above mentioned concept. For people counting at door locations, the Region of Interest (ROI) is manually marked around the door location (Figure 6.11 (left)). A pedestrian is tracked only in the ROI to build his trajectory. The trajectory represents the pedestrian motion within the ROI. The first coordinate in his trajectory represents the first time he was detected in the ROI and the last coordinate represents the last time he was detected when he is about to leave the ROI. As explained earlier, the difference between the last and first coordinates (that is the x coordinates difference and the y coordinates difference) in the pedestrian trajectory, is performed. The sign of the difference values of these $x$ and $y$ coordinates are utilized to estimate the direction of the pedestrian's motion using the conditions in algorithm 6.1. The conditions detect whether the pedestrian is going into the location or going out of the location.


Figure 6.11: Left - Region of Interest (ROI) for People counting. Right - People counting at door entrance for people entering (in count) and people exiting (out count).

Figure 6.11 (right) shows people counting at a library entry-exit point. The people counting results are displayed on the video stream along with the average waiting time. The average waiting time can be used to detect pedestrian flow bottlenecks (when waiting time is greater than 3 or 4 seconds) at the entry-exit points.

Five video clips are utilized to measure the performance of the proposed people counting method at door entrances. The videos are recorded from a library entrance location. The time duration of each video clip is 50 minutes. From Figure 6.12, it is clear that the proposed application is able to count almost all people who crossed the ROI.

## 6.2 Route Planner Web Application

Route Planner Web Application is a prototype solution to monitor and estimate the crowd level in the various modes of public transport (such as bus, train and taxis), so that commuters can make informed travel decisions. This application uses vision based techniques to estimate crowds at Mass Rapid Transit (MRT) train platforms, bus bays, inside MRT trains and taxi queues. It provides crowd information for all the modes of public transport. Videos monitoring a complete train platform, for a long period of time is required to test this solution. Such video data sets were not available. Hence, a virtual video, simulating the train platform was created to demonstrate the proof of concept of this solution. The solution explains the methods to estimate the crowd and is not intended to validate the pedestrian detection techniques accuracy.

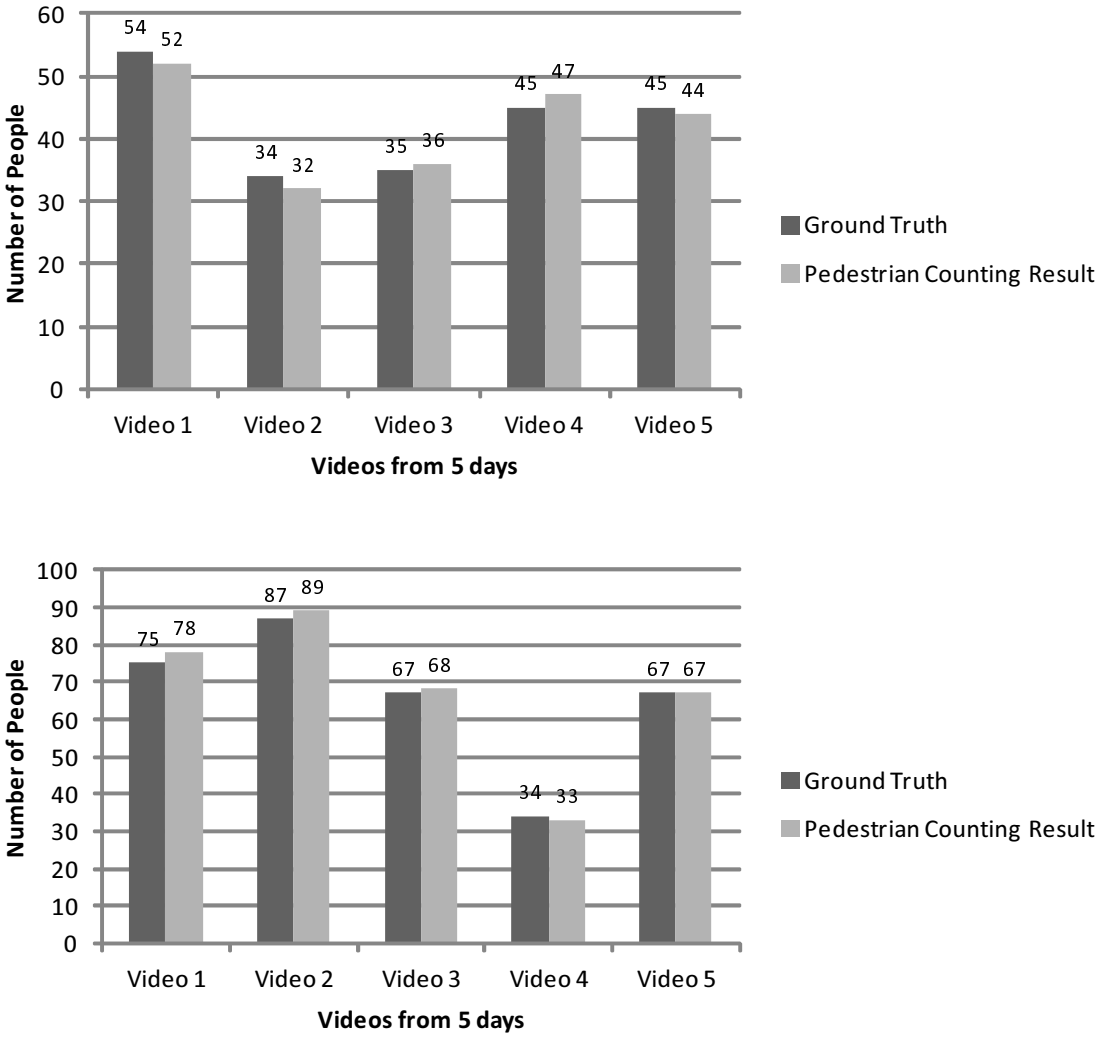



Figure 6.12: People counting results, in count (left) and out count (right) at library door entrance.

Figure 6.13 is a web interface of the application. The crowd information is shown on the route suggestions in textual and visual formats. The rain forecast information is fetched in real-time from the meteorological website and displayed along with the suggested routes.

When a commuter looks for route suggestions in Google maps, he or she can receive route suggestions together with the crowd level estimations along the routes and estimated traveling time (which includes the time delay due to the crowd). Information on weather forecast for open pavement areas along the suggested routes are also provided. The methods used to estimate the crowd level at the different locations are explained in the following section.

### 6.2.1 Crowd Estimation in Public Modes of Transport

The three main public modes of transport, namely, bus, train (Mass Rapid Transit (MRT)) and taxis, are considered in this section. The public modes of transport are quite different from each other in terms of the volume of traffic cleared, pedestrian occupancy area, frequency of operation, service start / end time and many other factors. Due to these factors, the crowd estimation solutions are different for the different modes of transport. Crowd estimation for the MRT train platforms is explained below after discussing about the MRT train platform simulated environment.

**1) Crowd Estimation at MRT Train Platforms** Due to unavailability of real-world MRT train platform videos, covering an entire platform for long periods of time from a surveillance point of view, a simulated environment of an MRT train platform is created using the PTV VisWalk software (explained in sub-section 6.1.1). The environment (Figure 6.14 (left)) is created with three pedestrian sources. The train platform is designed to be 150×5m, with the maximum possible crowd density of up to 3 people per square meter. The crowd density is set to random to resemble the real-world environment, where the crowd flow is uncontrolled. Pedestrian can enter the platform and can choose to wait near any entry-exit bay. The train schedule is fixed at one train every 3 minutes. So, the platform crowd increases after every train departure. In real-world train platforms with video surveillance, at least three cameras are used to monitor the entire platform. The

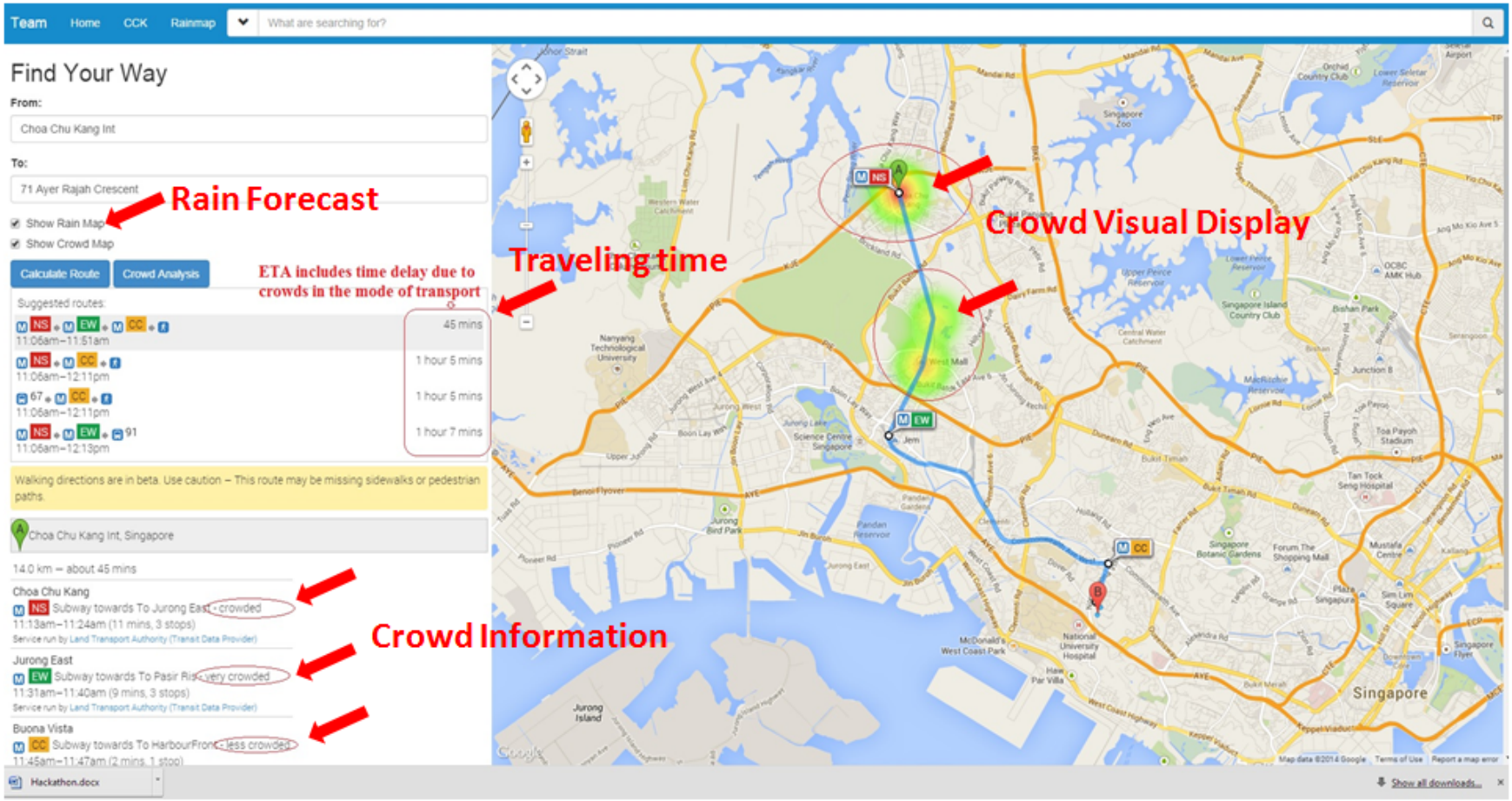



Figure 6.13: Route Planner Web Application. The crowd information is presented in textual format as well as in visual format (towards red color indicating high level crowd).

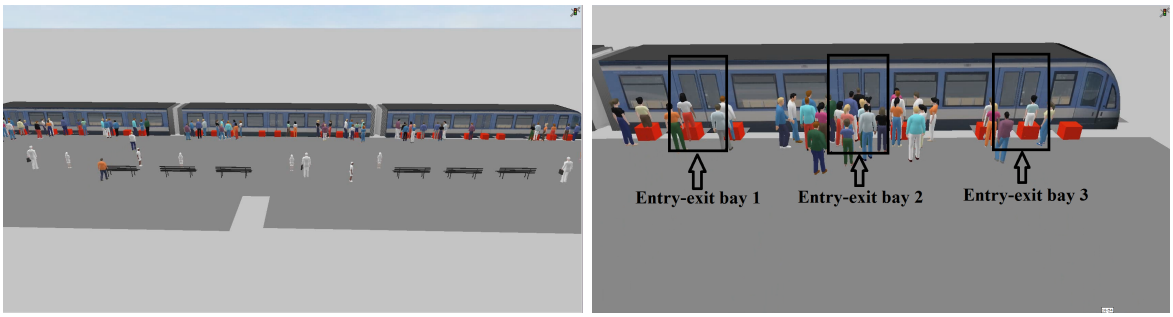



Figure 6.14: Mass Rapid Transit train platform environment (left), single car platform view (right).

developed environment is selected to have three view points, one for each passenger car's platform region. The view point for the last car region is shown in Figure 6.14 (right).

MRT train platforms tend to have high pedestrian occupancy compared to the long distance train platforms. MRT trains are extensively used by commuters in highly populous cities around the globe to reach their destinations quickly. The disruption of the MRT train services disrupt the commuter traffic on a large scale compared to that of the long distance trains. This is because the passenger traffic handled is very high compared to the long distance trains. Train commuters have a herding behavior [98], whereby they tend to crowd near the place where they enter the train platforms. Due to such behaviors, certain entry-exit bays remain empty, while others are overcrowded. People might go to the empty entry-exit bays, if they knew the location of empty bays before they came to the platform. The proposed approach for MRT platform crowd estimation provides this information to the commuters.

The stages in the proposed approach for MRT platform crowd estimation are shown in Figure 6.15. In this approach, the crowd density in the platform is estimated by adopting a spatial grid occupancy detection method. In this method, the entire platform is divided into a spatial grid ($G$). The grid structure is designed to have two rows and six columns for a single car platform region (shown in Figure 6.16), forming twelve grid cells. The number of columns is selected to be six to create two spatial grid cells per entry-exit bay and there are three such entry-exit bays per car in the simulated MRT environment which is shown in Figure 6.14 (right). It should be noted here that the first and the last spatial

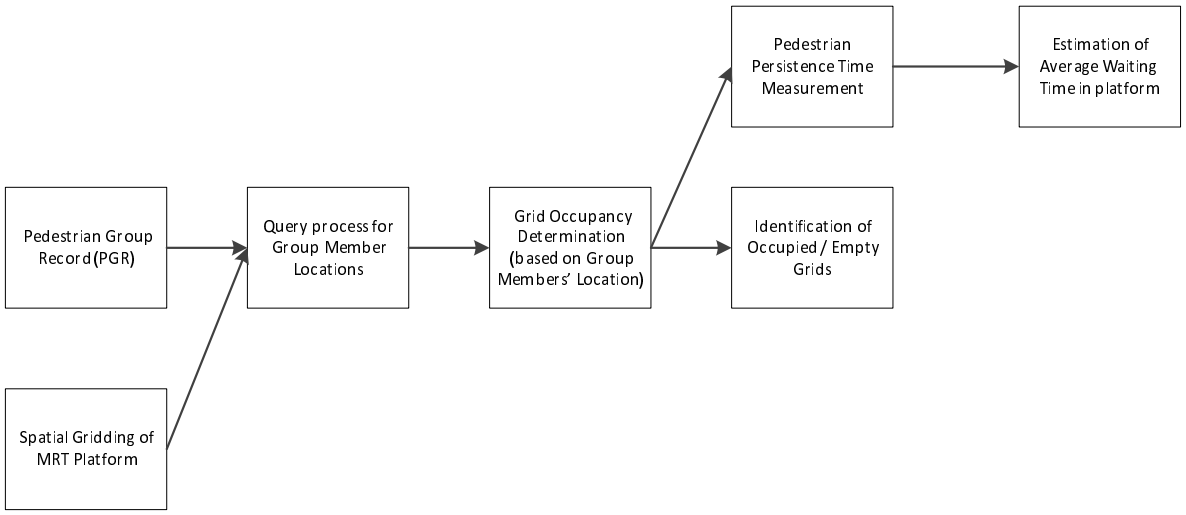



Figure 6.15: Stages in crowd estimation in MRT platforms.

grid cell are almost twice the size of the other grid cells. Such an uneven grid structure is employed to consider the extra space on the platform at the region of connection between the train cars and the extra space on the right side of the last car. $G$ is a spatial grid matrix shown in (6.1).

$$G = \begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} & G_{15} & G_{16} \\ G_{21} & G_{22} & G_{23} & G_{24} & G_{25} & G_{26} \end{bmatrix}, \tag{6.1}$$

where, G is the spatial grid matrix with cells, $G_{11}, ..., G_{26}$ storing the number pedestrian group members within these cells. The population of the cells in the grid matrix with the number of pedestrian group members is carried out using algorithm 6.2. The Pedestrian Group Record is queried at every video frame for the pedestrian groups. The latest coordinate in a group member's trajectory is considered as the current location of the group member. The pedestrian's location is checked for which grid cell he / she belongs to and the corresponding grid cell's value is incremented by 1. A similar process is performed for all the individual pedestrians.

After population of the cells (for a video frame), if a particular cell (for example, $G_{23}$) has the number of pedestrians greater than a user-defined grid cell capacity threshold value[2] (for example, 10), then the particular grid is considered to be crowded and

[2]The threshold should be selected based on the maximum capacity of a particular grid cell. For the

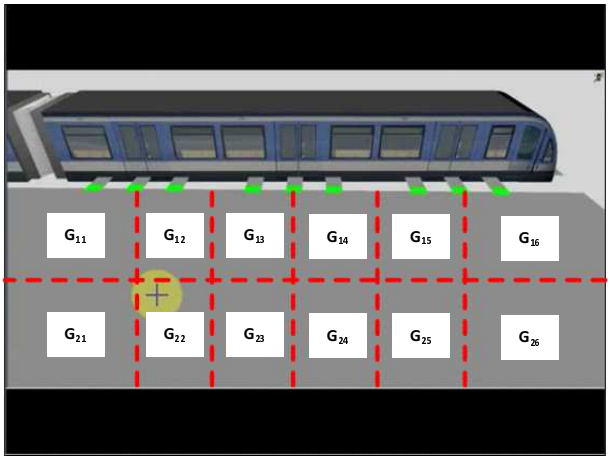



Figure 6.16: Spatial gridding of single car's train platform.

**Algorithm 6.2** Spatial grid matrix population.

*For every video frame*
| *Query Pedestrian Group Record for Pedestrian Groups*
| *For every Pedestrian Group*
| | *For every Group Member* "Check pedestrian's grid cell location with latest coordinate"
| | | *Check* $(x_{last}, y_{last})$ *is in which Grid Cell* $(G_{i,j})$
| | | $G_{i,j} = G_{i,j} + 1$; "Increment corresponding grid cell value by 1"
| | *end*
| *end*
*For every Individual Pedestrian* "Check pedestrian's grid cell location with latest coordinate"
| | *Check* $(x_{last}, y_{last})$ *is in which Grid Cell* $(G_{i,j})$
| | $G_{i,j} = G_{i,j} + 1$; "Increment corresponding grid cell value by 1"
| *end*
*end*

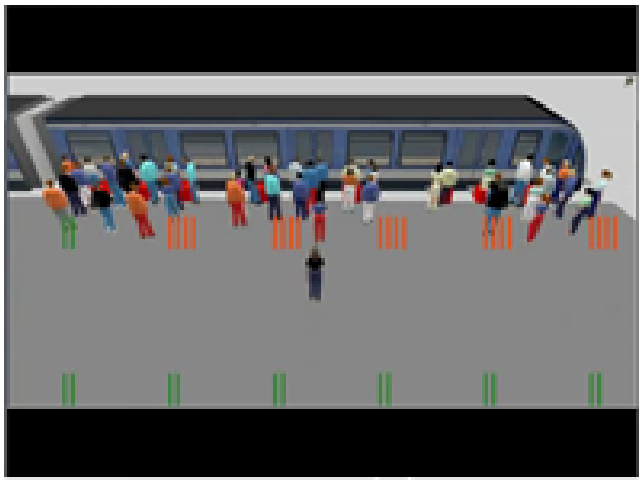


Figure 6.17: Spatial grid crowd estimation visualization. Red lines indicate crowded grid cells, green lines indicate not crowded or empty grid cells.

highlighted by red lines on the video stream. If the above mentioned condition is not satisfied, the cell is considered to be not crowded and highlighted by green lines on the video stream. The visual representation of the grid cell level crowd estimation is shown in the Figure 6.17. Pedestrian persistence times are measured from the time stamp feature (which includes the group start and end time) in the pedestrian group record and the average of the persistence time values is utilized as the average waiting time.

Figure 6.18 shows the grid occupancy measured (in terms of number of pedestrians) across three train arrival periods. Grid cell occupancy values are sampled once in thirty seconds. It is observed that the number of pedestrians peak at the last thirty seconds before a train arrival. This observation is consistent across all the grid cells which are adjacent to the platform ($G_{11}, G_{12}, G_{13}, G_{14}, G_{15}, G_{16}$) because people wait here to board the train. The remaining non-adjacent grid cells ($G_{21}, G_{22}, G_{23}, G_{24}, G_{25}, G_{26}$) have very few people compared to the adjacent grid cells.

**2) Crowd Estimation At Taxi Queues and Bus Bays** The crowd estimation at taxi queues and bus bays gives information on the queue status (like empty, partially full,

simulated train platform with a dimension of 150 × 5m, each small cell's area is 15.63 square meter and each big cell's area is 31.25 square meter. The maximum pedestrian capacity for a big cell is 93 pedestrian and for a small cell is 46 pedestrians.

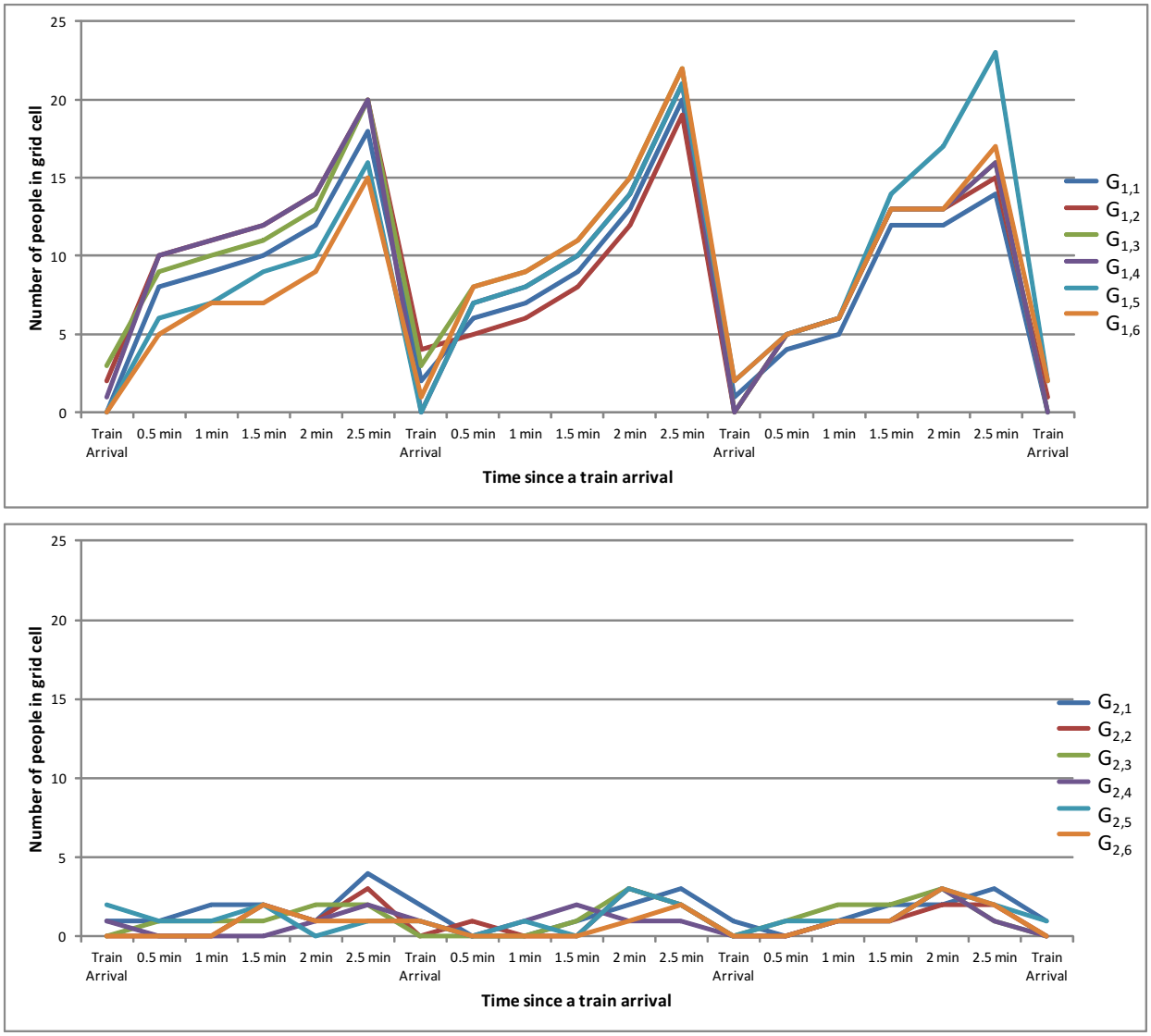



Figure 6.18: Grid occupancy measurement across three train arrival periods, for grid cell adjacent to the platform (top) and non-adjacent to the platform (bottom).

full) and the average waiting time. The solution for bus bays is similar in concept to the taxi queues (explained in Sec. 6.1.1).

## 6.3 Summary

Several applications were discussed in this chapter. They are people counting at different locations like queue regions, door entrances and crowd estimation at different public modes of transport. Certain simple, novel, learning methods are proposed: to automatically identify queuing regions and to estimate crowd levels in train platforms. All these applications are based on the information which is extracted and stored in the Pedestrian Group Record (PGR). PGR provides a simple and structured way of accessing important features of pedestrian group members. Thereby, the applications need not process the entire video archive to carry out their corresponding operations.

# Chapter 7

# Pedestrian Activity Prediction By Learning Pedestrian Motion Patterns

Pedestrians generally tend to visit public places to perform activities such as buying items, meeting friends, buying food at food outlets and many more. All these pedestrian activities tend to exhibit considerable continuity of motion. The corresponding motion patterns are similar for the same activity irrespective of the pedestrian. For example, pedestrians who are going to meet come closer irrespective of the pedestrians who are going to meet. The motion patterns are represented by the underlying motion parameter variations such as distance of travel, direction of travel and many more. Such motion parameter variations could be learned to predict pedestrian activity. This chapter explains a process to learn the motion parameter variations and discusses methods such as prediction of potential customer-approach to food outlets and prediction of possible future pedestrian groups. Supervised machine learning techniques such as Support Vector Machines are utilized to learn the motion parameter variations which lead to the pedestrian activity.

The process to learn the motion parameter variations are introduced in Sec. 7.1. An under sampling method to predict pedestrian activity several seconds ahead is also explained in the same section. The prediction of potential customer-approach to food outlets is explained in Sec. 7.2 along with the video data sets considered to evaluate the method. The last section explains the prediction of future pedestrian groups along with

the video data sets considered to evaluate the method.

## 7.1 Process to Learn Motion Parameter Variations

In this chapter, motion parameter variations are learnt using machine learning techniques to predict pedestrian activity. To predict pedestrians' activity, the motion parameter variations of the pedestrians are selected as the features to be learned. Depending on the pedestrian activity, either the individual motion parameters or the relative motion parameters are selected. Individual motion parameters represent every individual pedestrian's motion independent of the others. Some of the individual motion parameters are distance, speed and direction of travel. Relative motion parameters represent two or more pedestrians' motion collectively. Some of the relative motion parameters are relative distance, relative speed and relative direction of travel. In this thesis, individual motion parameters and pairwise (two pedestrians at a time) relative motion parameters are considered. For pedestrian activity involving more than one pedestrian such as pedestrians converging to form a group, it is intuitive to select the relative motion parameters rather than the individual motion parameters for learning.

The process of learning motion parameter variations is explained as follows. Pedestrians are detected and tracked to build their trajectories. The motion parameter variations are calculated from the pedestrian trajectories utilizing a Matlab script developed for this purpose. The Matlab script is explained in sub-section 7.1.1. The motion parameter variations which lead to the pedestrian activity and which does not lead to the activity are learnt using a supervised machine learning technique such as a Support Vector Machine (SVM). This is termed as the training phase. New pedestrians' motion parameter variations are checked against the learnt model to classify whether the new pedestrian is performing the same activity or not. The SVM learns the motion parameter variations and the ground truth on whether the corresponding pedestrian is performing the activity or not.

The motion parameters are fed as a feature vector to the SVM. The continuity of pedestrian motion is preserved by maintaining the temporal order of the feature vectors. The feature vector $\bar{x}^v$ at a video frame $v$ is

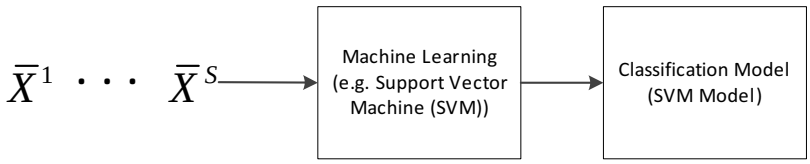



Figure 7.1: Support Vector Machine (SVM) training phase.

$$\bar{x}^v = \begin{bmatrix} x_1 & x_2 & x_3 & . & . & . & x_n & l \end{bmatrix}^T$$

where, $x_1$ to $x_n$ are the motion parameters considered as features to be learnt and $l$ is the label value. The label value represents the ground truth of the pedestrian activity. The ground truth is generated by a human observer who assigns a label value $l = 1$ as an element in a feature vector if the activity is performed or $l = 0$ if the activity is not performed. The feature vector represents individual pedestrians, if the individual motion parameters are learnt. The feature vector represents a pair of pedestrians, if the relative motion parameters are learnt.

To train the SVM, samples of pedestrian motion parameter variations which lead to the pedestrian activity (positive samples) and samples which do not lead to the pedestrian activity (negative samples), need to be collected from videos. Each sample is a sequence of feature vectors. Each video has numerous samples ($s$), where $s = 1...S$ . Each sample is represented by a feature vector set $\bar{X}$, consisting of feature vectors from every video frame. The feature vector set $\bar{X}$ is

$$\bar{X} = \begin{bmatrix} \bar{x}^1 & \bar{x}^2 & \bar{x}^3 & . & . & . & \bar{x}^V \end{bmatrix}$$

where, $v = 1, 2, ...V$, $V$ is the number of video frames in a sample.

Figure 7.1 shows the training phase where the collected feature vector sets $\bar{X}^1$ to $\bar{X}^S$ are fed as input to the SVM to build an SVM model. This completes the training phase. The SVM model is used to classify new feature vectors which do not have labels. The SVM classification output can be interpreted as a prediction because the classification output represents the outcome of the pedestrian motion in the future as the labels are for an event (i.e. the outcome) which happens later in the future.

Feature vectors are generated from every video frame. If all the feature vectors are fed in the temporal order to the SVM, the classification output from the learnt SVM kernel is a predicted outcome for the next frame. This is a trivial prediction. To predict an outcome several seconds ahead, for example, two seconds ahead, feature vectors of only few (for example three) video frames per two second time intervals are considered for SVM learning. In this example, the sampling interval of the learnt temporal sequence is 2 seconds. So, the predicted outcome of the SVM model will be for 2 seconds ahead. This type of under sampling method is adopted in the two prediction methods discussed in the following sections.

Training and classification of new feature vectors are performed using an open source software termed Weka [99]. The process to generate the ground truth for the motion parameter variations is explained in the following sub-section.

Several training methods such as linear SVM, non-linear SVM -Radial Basis Function (RBF), SVM with meta-algorithms such as AdaBoost [27], Decision stump classifier [100], Neural Network [101] and Random Forest [102] were also tested to learn the motion parameters. All these standard machine learning techniques were tried, but their results are not shared in this chapter as they were of almost the same accuracy as that of the linear SVM. The results of the linear SVM are alone shared in this chapter as it is observed that linear SVM is able to achieve almost the same level of accuracy as the other complex techniques such as non-linear SVM, Neural Network and others. Also the linear SVM takes a lesser training time. The results of the other machine learning techniques are shared in Appendix C. The training time measurements are also shared in Appendix C.

### 7.1.1 Motion Parameter Annotation with Ground Truth

The video data sets considered for analysis have a video frame rate of 25 frames per second. The video data sets are processed to extract samples of pedestrian activity, which form the positive samples. Equal number of samples which does not involve the pedestrian activity are also extracted, which form the negative samples.

A Matlab script is developed to manually annotate the trajectories with labels (0 - will perform the activity, 1 - will not perform the activity). The pedestrian trajectories are

fed to the script to generate the motion parameter sequences (for example, the distance of travel, relative distance of travel, relative direction of travel and so on). The Matlab script consists of a set of formulas to calculate the motion parameters iteratively, for all the instances in the pedestrian trajectories. The script allows the labeler to feed in the label for the samples. By running the Matlab script, the samples are generated. The next section explains the prediction of potential customer-approach to food outlet by learning the individual motion parameters.

## 7.2 Prediction of Potential Customer-Approach to Food Outlets

A pedestrian's approach to a food outlet is predicted by learning the individual motion parameter variations. Since the pedestrian activity predicted is an individual pedestrian's activity, the individual motion parameters are selected as the features to be learnt. In this problem, the possible outcomes of a pedestrian's motion are limited to two: pedestrian going to a food outlet (positive), pedestrian passing by or leaving a food outlet (negative). As explained in Sec. 7.1, a Support Vector Machine (SVM) is utilized to learn the motion parameter variations which lead to pedestrians reaching a food outlet. The motion parameter variations are fed as a sequence of feature vectors, preserving the temporal order of the feature vectors.

Two sets of individual motion parameters are utilized for SVM learning. They are the basic set and the extended set of individual motion parameters. The basic set of individual motion parameters includes,

$x_1$ - Distance $s_\epsilon$

$x_2$ - Velocity $\bar{v} = \frac{\triangle \bar{s_\epsilon}}{\triangle t}$, where $\bar{s_\epsilon}$ is the displacement, $\triangle$is a change in the corresponding physical quantity, $t$ is time.

The extended set of individual motion parameters include,

$x_1$ - Distance $s_\epsilon$

$x_2$ - Direction $dir\,\epsilon\,(1, 2, 3, 4, 5)$

$x_3$ - Speed $\nu = \frac{\triangle s_\epsilon}{\triangle t}$

$x_2$ - Velocity $\bar{v} = \frac{\triangle \bar{s_\epsilon}}{\triangle t}$, where $\bar{s_\epsilon}$ is the displacement.

$x_5$ - Acceleration $\bar{a} = \frac{\triangle \bar{v}}{\triangle t}$

$x_6$ - Gradient of speed $\nabla\nu = \frac{\partial \nu}{\partial x}i + \frac{\partial \nu}{\partial x}j$, with respect to two dimensional x, y coordinates and i, j are standard unit vectors.

$x_7$ - Instantaneous velocity $\bar{v}_{inst} = \frac{\triangle \bar{r_\epsilon}}{\triangle t}$, where $\bar{r_\epsilon}$ is the displacement in one time unit.

$x_8$ - Rate of change of direction $R(dir) = \frac{\triangle dir}{\triangle t}$, where $R$ is used to represent rate of change of a quantity.

$x_9$ - Instantaneous speed $\nu_{inst} = |\bar{v}_{inst}|$

$x_{10}$ - Gradient of instantaneous speed $\nabla\nu_{inst} = \frac{\partial \nu_{inst}}{\partial x}i + \frac{\partial \nu_{inst}}{\partial x}j$

Here, the direction of travel is quantized to five directions. The pedestrian's current location with respect to the location in the previous video frame is represented as the direction of travel. The difference of the x coordinates and the y coordinates between the pedestrian's current and previous locations are utilized to determine pedestrian's motion direction. The five motion directions represent pedestrian moving north or south of the previous location, moving east or west of the previous location, moving northeast or southwest of the previous location, moving southeast or northwest of the previous location and pedestrian remaining stationary. The motion directions are quantized in this manner to easily encode the motion direction. The five motion directions are shown in Figure 7.2. In the extended set of individual motion parameters, $x_1$ to $x_5$ are the basic motion parameters which describe a pedestrian motion. Features $x_6$ to $x_{10}$ measure the variations in the basic motion parameters. Hence, all these features which describe the pedestrian motion are selected for SVM training.

Videos from five different food outlets are utilized to test the potential customer-approach prediction method. The video data sets and the samples recorded from these video data sets are explained in the next sub-section.

### 7.2.1 Video Data sets

The positive and the negative samples are extracted from video recordings of food outlets in National University of Singapore. Figures 7.3, 7.4 and 7.5 shows the five food outlets considered, along with the green colored positive samples and the red colored

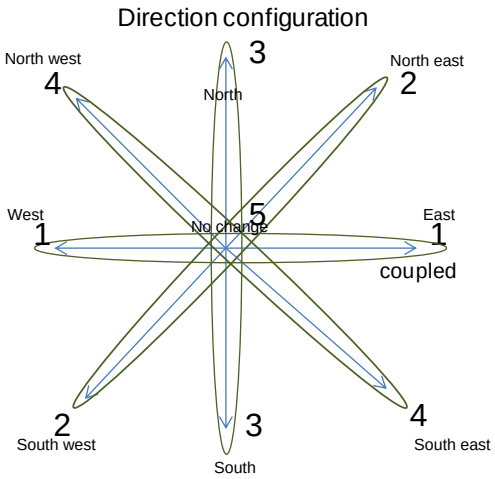



Figure 7.2: Discrete directions of travel.

Table 7.1: Ground truth of the video data sets.

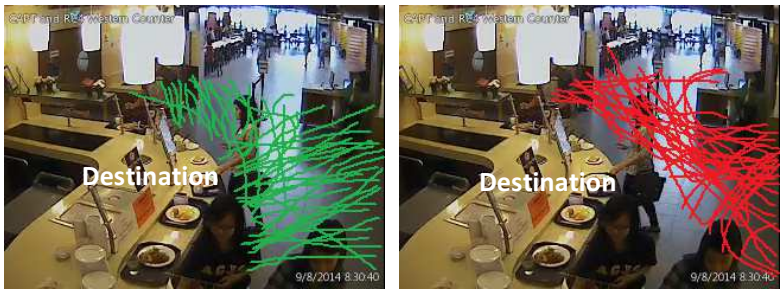
| Food outlet | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Positive samples | 623 | 752 | 768 | 880 | 977 |
| Negative samples | 704 | 723 | 753 | 854 | 916 |

negative samples. Table 7.1 lists the number of positive and negative samples extracted from these video data sets.

### 7.2.2 Prediction Performance Evaluation

Three experiments are performed to build and analyze the potential customer-approach prediction method. They are as follows.

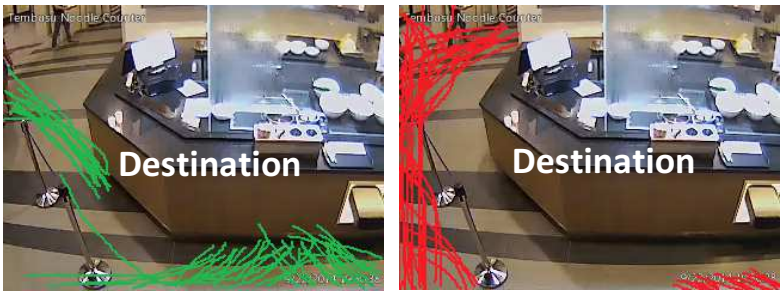
**Experiment 1** This experiment is performed to find which kernel type of SVM is suitable for predicting the pedestrian's destination. Linear [42] and Radial Basis Function (RBF) [103] kernels are tried in this experiment. AdaBoost meta algorithm [27] is tried to improve the prediction accuracy. From Table 7.2, it is evident that the linear kernel with AdaBoost routine utilizing the extended set of motion parameters performs better than the other configurations. This configuration exhibits the best possible prediction accuracy.




(a) Food outlet 1, positive (left) and negative (right) samples.


(b) Food outlet 2, positive (left) and negative (right) samples.

Figure 7.3: Food outlet locations with pedestrian motion samples.




(a) Food outlet 3, positive (left) and negative (right) samples.


(b) Food outlet 4, positive (left) and negative (right) samples.

Figure 7.4: Food outlet locations with pedestrian motion samples. Continued.

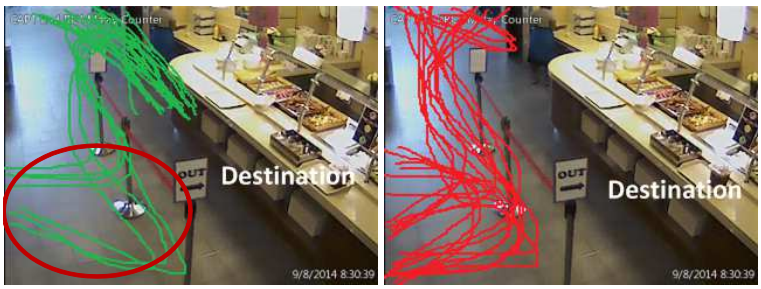



Figure 7.5: Food outlet 5, positive (left) and negative (right) samples. Continued. Here, the positive samples are similar to the negative samples in the initial part (indicated by red oval).

Table 7.2: Prediction Performance with Linear and Radial Basis Function (RBF) Kernel. A - With AdaBoost meta algorithm routine. Basic, Extended - refers to the motion parameters (features) for the SVM learning.

| Food Outlet | Motion Parameters | Kernel Type | | | |
|---|---|---|---|---|---|
| | | Linear | Linear (A) | RBF | RBF (A) |
| 1 | Basic | 62 | 64 | 63 | 64 |
| | Extended | 83 | 85 | 68 | 69 |
| 2 | Basic | 65 | 68 | 60 | 62 |
| | Extended | 86 | 89 | 69 | 68 |
| 3 | Basic | 62 | 65 | 58 | 59 |
| | Extended | 83 | 87 | 66 | 66 |
| 4 | Basic | 61 | 65 | 58 | 58 |
| | Extended | 82 | 86 | 67 | 65 |
| 5 | Basic | 54 | 58 | 52 | 51 |
| | Extended | 75 | 80 | 60 | 57 |

Hence, the linear kernel with AdaBoost routine is selected for pedestrian destination prediction. Experiment 1 is performed by considering the feature vectors generated from all the video frames (25 frames for every second) for SVM learning. The prediction accuracies are mostly more than 60 per cent because the motion parameter variations associated with the pedestrian motion patterns are continuous and similar within the sample groups (positive and negative sample groups).

**Experiment 2** This experiment is performed to find the minimum number of video frames[1] required (for SVM learning) in a second to achieve an acceptable prediction accuracy which is close to the best possible prediction accuracy. The under sampling method discussed in the last part of Sec. 7.1 is adopted to predict one second ahead. So, the sampling interval of this experiment is one second and the prediction depicts the outcome in the next second. The best possible prediction accuracy is achieved when all

[1]Here, video frames refer to the feature vectors generated from the video frames.

Table 7.3: Prediction Performance (in %) at different video frame sampling rates.

| Under sampling rate (video frames per second) | Food outlet | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| one | 68 | 69 | 66 | 60 | 52 |
| two | 73 | 75 | 72 | 70 | 64 |
| three | 80 | 84 | 82 | 81 | 74 |
| four | 81 | 85 | 82 | 82 | 75 |
| five | 82 | 86 | 84 | 82 | 75 |
| six | 83 | 88 | 86 | 83 | 78 |
| twenty five | 85 | 89 | 87 | 86 | 80 |

the 25 video frames per second are available to the SVM. The best possible prediction accuracy is outlined in experiment 1.

The results of this experiment are shown in Table 7.3. With increasing number of video frames (i.e. feature vectors) per second, the prediction accuracy approaches the best possible prediction accuracy. The best possible prediction accuracy is achieved when all the 25 video frames are available to the SVM. From the experiment, the prediction accuracy for six video frames per second is close to the best possible prediction accuracy. Under sampling rate of three is selected and not six. Ensuring six video frames per second is a high requirement and the performance difference between size three and six is not large (approximately 3%). Based on experience with camera surveillance systems, it is possible to consistently extract three video frames per second from the camera-server network.

**Experiment 3** Experiment 3 is performed to determine how far ahead an outcome prediction can be made with an acceptable prediction accuracy. This experiment is an extension of Experiment 2. Different sampling intervals are tried (three video frames every 1 second interval, three video frames every 2 second interval and so on).

It is clear from the Table 7.4 that the prediction accuracy decreases with increasing sampling interval size. This is because the motion parameter variations are not completely

available to the SVM at larger sampling interval sizes. The continuity of motion is also lost when a large sampling interval size is employed because the motion parameter variations associated with the neglected (due to the large sampling interval sizes) video frames are lost. The misclassifcation of the positive feature vectors is more than the negative feature vectors. The rate of misclassifaction of positive feature vectors increases with an increase in the sampling interval size. Typically, a positive sample's (a pedestrian going to the destination) duration is 3 to 4 seconds. If only three video frames' feature vectors represent this sample in the SVM model (for 'four seconds' and larger sampling interval sizes), the prediction performance is drastically reduced as the corresponding positive feature vectors are misclassified. While the negative samples' (pedestrians leaving the destination or passing by the destination) duration is more than 5 seconds. So, more number of video frames' feature vectors represent the negative samples in the SVM model and hence, the prediction performance for the negative samples is not deteriorating as much as the positive samples' prediction performance.

For the food outlet 5, the misclassification of the positive feature vectors are higher than that of the other food outlets. This is because the motion parameter variations in the positive samples are similar to that of the negative samples. This observation is highlighted by a red oval in the negative samples in Figure 7.5. For potential customer-approach prediction at the food outlets considered, a maximum of 'three seconds ahead' prediction can be made with an acceptable prediction accuracy.

## 7.3 Prediction of Future Pedestrian Groups

In the physical world, pedestrians tend to come closer together to interact. Such convergence of pedestrians, mostly leads to a pedestrian group formation. Several relative motion parameters may vary when the pedestrians converge, such as relative distance between pedestrians, relative direction of travel and many more. Pedestrian relative motion parameter variations which lead to a pedestrian group formation, are hypothesized as a sufficient condition to predict possible future pedestrian groups. The hypothesis is tested using real-world surveillance video feed. The method to predict future groups is explained below.

Table 7.4: Prediction Performance (in %) at different temporal resolutions. In this table, PA - Prediction Accuracy, MC - Miss-Classification, +ve - positive samples, -ve - negative samples. Different temporal resolutions are tried. If the temporal resolution is one second (one feature vector block input in every one second interval to the SVM), the prediction depicts the outcome in the next second.

| Food outlet | 1 | | | 2 | | | 3 | | | 4 | | | 5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sampling interval (three video frames per "x" seconds) | PA | MC | | PA | MC | | PA | MC | | PA | MC | | PA | MC | |
| | | +ve | -ve | | +ve | -ve | | +ve | -ve | | +ve | -ve | | +ve | -ve |
| x = 1 | 81 | 12 | 6 | 84 | 10 | 4 | 82 | 12 | 4 | 81 | 14 | 3 | 74 | 21 | 4 |
| x = 2 | 76 | 16 | 7 | 77 | 16 | 6 | 75 | 17 | 7 | 73 | 20 | 6 | 64 | 27 | 8 |
| x = 3 | 72 | 18 | 9 | 72 | 19 | 8 | 69 | 21 | 8 | 67 | 23 | 9 | 58 | 31 | 10 |
| x = 4 | 58 | 31 | 10 | 58 | 30 | 11 | 55 | 32 | 12 | 52 | 35 | 11 | 45 | 42 | 12 |
| x = 5 | 49 | 37 | 13 | 48 | 37 | 14 | 45 | 38 | 16 | 40 | 40 | 19 | 32 | 32 | 35 |

The pedestrians' activity of converging to form a group is predicted by learning the relative motion parameter variations between the pedestrians. Since the pedestrian activity predicted involves motion between two or more pedestrians, the relative motion parameters are selected as the features to learn. In this work, the possible outcomes of pedestrians' motion are limited to two: pedestrians converging to form a group (positive), pedestrian not converging (negative). A Support Vector Machine (SVM) is utilized to learn the motion parameter variations which lead to pedestrians forming groups. The motion parameter variations are fed as a sequence of feature vectors, preserving the temporal order of the feature vectors.

The relative motion parameters selected for SVM learning are,

$x_1$ - Relative distance $s_{\epsilon_{ij}} = s_{\epsilon_i} - s_{\epsilon_j}$

$x_2$ - Relative velocity $\bar{v}_{ij} = \bar{v}_i - \bar{v}_j$

$x_3$ - Relative acceleration $\bar{a}_{ij} = \bar{a}_i - \bar{a}_j$

$x_4$ - Relative direction $dir_{ij}$

$x_5$ - Instantaneous relative velocity $\bar{v}_{inst_{ij}} = \frac{\triangle \bar{r}_{\epsilon_{ij}}}{\triangle t}$, where $\bar{r}_{\epsilon_{ij}}$ is relative displacement between pedestrian i, j in one time unit.

$x_6$ - Rate of change of instantaneous relative velocity $R(\bar{v}_{inst_{ij}}) = \frac{\triangle \bar{v}_{inst_{ij}}}{\triangle t}$ , where $R$ is used to represent rate of change of a physical quantity.

$x_7$ - Rate of change of relative direction $R(dir_{ij}) = \frac{\triangle dir_{ij}}{\triangle t}$

$x_8$ - Rate of change of relative acceleration $R(\bar{a}_{ij}) = \frac{\triangle \bar{a}_{ij}}{\triangle t}$

Here, the relative direction is calculated in the same manner as explained in Sec. 7.2. The only difference is that the x and y coordinate differences are calculated between two pedestrian's current location instead of a single pedestrian's current and previous locations. In this list of features, $x_1$ to $x_4$ are the basic relative motion parameters which describe the relative motion between the pedestrians. Features $x_5$ to $x_8$ measure the variations in the basic relative motion parameters.

The major differences between this prediction method and the prediction method discussed in Sec. 7.2 are explained as follows. The relative motion parameters are selected as the features to learn in this prediction method while in potential customer-approach prediction, the individual motion parameter variations are learnt. In this prediction method, every feature vector set $\bar{X}$ consists of feature vectors for all pairwise combinations of pedes-

Table 7.5: Group formation ground truth for data sets 1 to 6. Here, positive samples - group formation samples, negative samples - samples with no group formations.

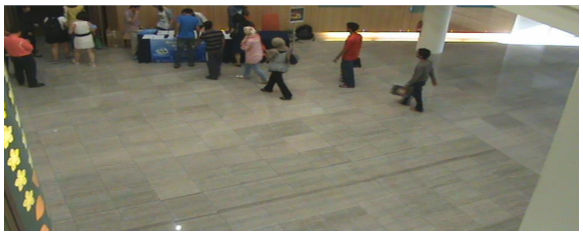
| Data set | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Positive samples | 9 | 23 | 9 | 8 | 32 | 4 |
| Negative samples | 12 | 17 | 8 | 2 | 16 | 6 |

trians. The Matlab script utilized considers all the pairwise pedestrian combinations to calculate the relative motion parameters. For example, if there are $M$ pedestrians in a sample, $Pr$ pairwise combinations of pedestrians are considered by the Matlab script to generate the feature vectors. Here, $Pr = {}^{M}C_2$ .

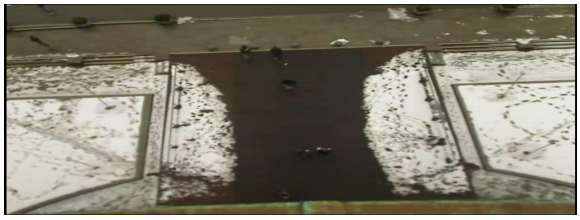
Six different video data sets are utilized to test the future pedestrian group prediction method. The video data sets and the samples recorded from these video data sets are explained in the next sub-section.

### 7.3.1 Video Data sets

The positive and the negative samples are extracted from six different video data sets. Figure 7.6 provides a snapshot of all the video data sets used for the future group prediction. The data sets are recorded from various view points and various locations. Video data sets with different view points and locations is selected to test whether the learning from one view point is applicable in another view point. The overall objective is to find whether the motion parameter variations which lead pedestrians converging to form groups can be generalized for all the locations considered. Data set 1 which is the NUS Movie Event (NUSME) data set, is already discussed in Chapter 4. Data set 2 [2] is recorded from a top angle with almost no pedestrian occlusions. Data set 3 from the town center video database [104] records the pedestrian movement in a town center, from an oblique camera angle. Data set 4, 5 and 6 were recorded in a library at the National University of Singapore. Table 7.5 lists the number of positive and negative samples extracted from these video data sets.



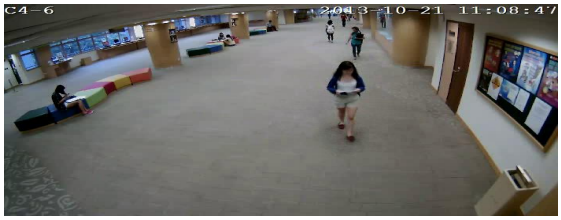
Data set 1
Camera angle 27 degree



Data set 2
Camera angle 0 degree

Data set 3
Camera angle 57 degree


Data set 4
Camera angle 35 degree




Data set 5
Camera angle 22 degree


Data set 6
Camera angle 22 degree

Figure 7.6: Data sets used for future pedestrian group prediction.

### 7.3.2 Prediction Performance Evaluation

The linear Support Vector Machine (SVM) [42] is utilized to learn the relative motion parameter variations and build the SVM model. For performance evaluation, the SVM model learnt from each data set is applied to the data sets including the same data set. The percentage split approach [105] is adopted when learning and testing is performed on the same data set. The percentage split approach divides the feature vector samples in a data set into two sets without shuffling the feature vectors in individual samples - one for training and the other for testing. A ratio of 60 percent of feature vector samples for training and the remaining 40 percent for testing is adopted. When the SVM model is applied to other data sets, the entire set of feature vectors of a selected data set is utilized. The temporal order of the feature vectors is not altered. As explained in Sec. 7.1, the feature vector sets are under sampled and, thereby, their predictions depict the pedestrian's group state at several seconds ahead. Feature vectors from only three video frames at two second (i.e. 50 video frames for a 25 frames per second video) sampling intervals are selected for the feature vector set. Hence, the sampling interval size is two seconds and the prediction performed with such a sampling interval depicts the pedestrians' state (will converge to form a group or not) at two seconds ahead. The results shared in this sub-section are for two seconds ahead predictions.

The future group prediction accuracy is visually represented as a confusion matrix (refer Figure 7.7). In this representation, each row represents a single data set selected as a training data set and the learning is applied to all the data sets. Each cell in the representation is a test trial on a data set. For example, the cell represented by second row, third column, shows the prediction accuracy of the SVM model built from data set 2 tested on data set 3. The heat map indicates the percentage of prediction accuracy. A value of 1 indicates 100 percent prediction accuracy, which is color coded as dark red.

The prediction accuracy across the data sets are above 60 percent most of the times. This considerable performance is explained as follows. In the video data sets, pedestrians are observed to converge to form a group. The convergence motion is generally continuous without major deviations from the pedestrians' path. Such a convergence occurs over a period of time representing the positive samples. The positive samples occur for a time

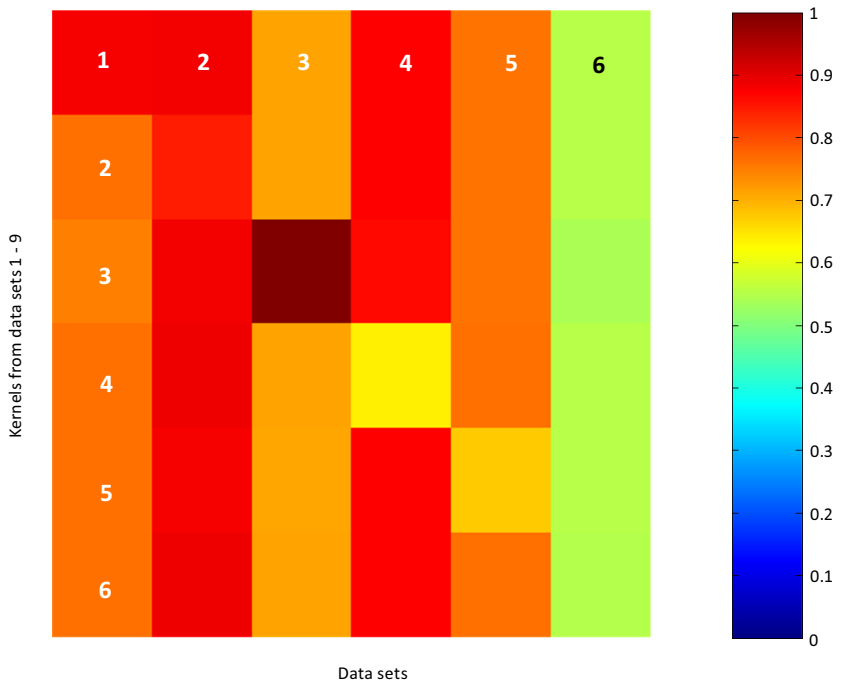



Figure 7.7: Prediction performance for data sets 1 to 6. The heat map indicates percentage of prediction accuracy. A value of 1 (represented by dark red color) indicates 100% accuracy.

duration of six seconds on an average in the considered data sets. The motion parameter variations associated with such positive samples are different from the negative samples which do not lead to pedestrians converging. Hence, the pedestrian activity of converging to group and not converging are considered to be mutually exclusive in terms of the associated motion parameter variations. This mutual exclusiveness is the reason that the SVM model is actually learning and not classifying randomly. The good prediction performance exhibited when the SVM model from one data set is applied to other data sets (cross data set validation) is considered as a proof for the SVM learning of motion parameter variations. There is another probable reason for the good performance and is explained as follows. The labeling was performed by humans who make judgements based on their intelligence. For example, let us consider that an observer labels two pedestrians to form a group for a sample in a data set. He / She labels not only based on the pedestrians' spatial proximity, but also based on their gestures and other non-verbal cues (which they see from the video data set). Such an intelligent labeling is associated with the motion parameter variations which happened during that period of time (before they form the group). Even though the visual cues such as gestures are not available explicitly to the proposed method of predicting future pedestrian groups, they are indirectly learnt by the SVM by learning the motion parameter variations associated with these visual cues.

The prediction accuracy of the data set 6 is lower compared to the prediction accuracy of the other data sets. The low performance is explained as follows. Data set 6 is observed to have motion parameter variations which are not continuous for more than three seconds. This behavior is attributed to the pedestrian's short period (less than three seconds) of convergence to form groups. While in the other data sets, the pedestrians converge over a period of time (average of six seconds) exhibiting longer periods of continuity of motions. Hence, the motion parameter variations learnt in data set 6 is considered to be different from that of the other data sets.

## 7.4 Summary

This chapter highlighted two methods which predict pedestrian activities by learning pedestrian motion patterns. They are potential customer-approach prediction and prediction of possible future pedestrian groups. Potential customer-approach prediction by learning the individual motion parameter variations was demonstrated with the food outlets example. The pedestrians' approach to the food outlet is predicted with considerable accuracy of greater than 60 percent most of the times. Future groups were predicted with considerable accuracy of greater than 60 percent. The SVM model learnt from one location could be applied to the other locations which have comparable crowd densities and considerable uniformity in pedestrian motion.

# Chapter 8

# Conclusion

In this thesis, several algorithms to understand pedestrian activity based on their motion parameters are presented. The work focused on identifying and predicting the pedestrians' group and individual behavior. The algorithms' efficiency in identifying such behavior was highlighted using simulated and real-world video feeds. Some of the applications developed using the algorithms, have been deployed in real-world scenarios.

## Summary of Contributions

A summary of the major contributions are presented in this section.

**Automatic pedestrian group identification** A method to automatically identify pedestrian groups is presented. This method utilizes a novel Non-recursive Motion Similarity Clustering (NMSC) algorithm to cluster pedestrians based on their motion similarities. This method is inspired by social psychological principles on group behavior [17]. In the NMSC algorithm, individual pedestrians are detected and tracked in video scenes. Pedestrians are automatically clustered based on their pairwise motion similarity by considering their relative distance, relative speed and relative direction of motion. Pedestrian clusters which persist for a period of time are identified as pedestrian groups.

Performance evaluation against existing related works reveal that the proposed method identifies pedestrian groups with better accuracy and can identify large groups with more than three pedestrians (which existing related works cannot identify).

**Pedestrian Group Record** Several features are extracted from the process of pedestrian group identification, namely group size, group member identity, group members' trajectory, detection count, absence count and time stamp at which the group is identified. These features are stored in a data structure termed Pedestrian Group Record (PGR). Several applications are based on the features which are stored in the PGR.

**Pedestrian group record visualizations** Visualizations of spatial and temporal pedestrian information (from the pedestrian group record) are presented. The Pedestrian Groups visualization helps to monitor the pedestrian events in real-time. The Group Membership History visualization provides a clear picture of the history of pedestrian events that happened without the need to search the entire video. The Pedestrian Spatial distribution visualization identifies occupancy (popular) areas within a monitored region. These visualizations are demonstrated by real-world cases in the thesis.

**Real-time pedestrian visits and meetings identification system** A system to automatically identify pedestrian meetings and visits from surveillance videos is presented. The real-time system has a Pedestrian Detection and Tracking module, Pedestrian Group Identification module (NMSC algorithm), a Pedestrian Group Record (PGR) and three visualizations of the PGR information. The system uses the information in the PGR to identify the pedestrian meeting and split events. A meeting event is identified when a newly formed group's existence is confirmed by checking in the pedestrian group record. A split event is identified when a group's termination is confirmed by checking in the pedestrian group record. The real-time system has been deployed in residential halls in National University of Singapore (NUS) after performing calibrations to address the issue of loss of video frames in the internet transmission medium. The system is found to have an event identification accuracy of greater than 80 percent.

**Applications based on pedestrian detection and pedestrian group identification** Applications based on pedestrian detection and pedestrian group identification are presented. These include: people counting at queue regions, people counting at door entrances and crowd estimation at different public modes of transport. Certain simple, novel, learning methods are proposed: to automatically detect queuing regions and to estimate crowd levels in train platforms. All these applications are based on the information which is extracted and stored in the Pedestrian Group Record (PGR).

**Learning motion parameter variations to predict approach of potential customers and possible future pedestrian groups** A process to learn motion parameter variations (for example distance, speed, direction of travel) which lead to a pedestrian activity such as an approach of potential customers to food outlets is presented. Supervised machine learning techniques such as Support Vector Machines are utilized to learn the motion parameter variations which lead to the pedestrian activity. Human annotated labels (0 indicating pedestrian activity performed , 1 indicating pedestrian activity not performed) are utilized during the process of learning to build the SVM model. Motion parameter variations of new pedestrians are classified using the SVM model. This process of learning motion parameter variations is adopted in two proposed methods to predict approach of potential customers to food outlets and to predict possible future pedestrian groups. The These methods can predict the pedestrian's status ahead of time with an acceptable prediction accuracy.

# Future Directions

**Tracking pedestrian groups across different regions** Identifying pedestrians in groups by employing facial recognition will help to continue tracking pedestrians who move from one region to another. There are standard face recognition and tracking techniques which could be used for tracking pedestrians across regions. Such a tracking mechanism will ensure continuous surveillance of the pedestrians of interest and helps in building a pedestrian group record (which is not limited by a single camera's region).

**Security in public locations** The relevance of the discussed system and its information about pedestrian meetings can be explored in different scenarios, which can provide insights on how pedestrians behave in different environments. In scenarios such as surveillance applications, the automatic detection of build-up of unusually large groups of people can alert security personnel of impending issues.

**Group configuration analysis** The current work identifies groups of arbitrary size in pedestrian movement. The effect of stationary obstacles and group size on the group member's spatial structure (group configuration) can be studied. The study of group behavior is not only of interest for sociologists but also of importance for realistic crowd simulation and evacuation planning. For example, researchers have shown that pedestrian behavior models learned from video observations can be useful for tracking [106, 107, 108, 109] and activity recognition [110]. The prediction of pedestrian motion is usually determined by a repulsive force field [111], and it has been reported that a primary failure mode of this model is when groups of pedestrians walk together [107]. This finding suggests that a model component should be added to take group configurations into account. Modeling for evacuation planning requires such group configuration components to accurately model the evacuation times for different group sizes. Group configuration models can be built by studying pedestrian movement patterns in different groups (i.e. different group sizes and group types - a family, group of friends) from real world observations.

**Group interaction analysis** Motion of groups is not only affected by the stationary obstacles, but also by other groups and individuals in the vicinity [13, 15]. Crowd behavior analysis in situations of mob, needs an understanding of the dynamic interaction between groups. Group Interaction analysis can identify the aggressors in a mob situation and thereby, helping the authorities in charge to take prompt action. Video data sets of real world mob situations can be identified and group movement patterns and their correlation with each other can be determined. A model to predict the movement of groups of different size in interaction with other groups can be built based on group movement pattern and their correlations.

**Motion parameters as scene descriptors** Motion parameters and its thresholds are selected to define and identify pedestrian groups. Every type of scene (e.g. a mall, an exhibition, a football match, immigration queue) has its own range of values (for typical activities) for the motion parameters. So selection of motion parameters and it's thresholds is a non-trivial task and the selected motion parameters might hold crucial information about the scene under study. Techniques to identify the dominant motion parameters of a scene (e.g. velocity is expected to vary more than the other motion parameters in sports activity like football) can be developed. The dominant motion parameters might be an indirect cue for automatic identification of the type of scene being analyzed.

# Bibliography

[1] Dalal N.and Triggs B. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, 1:886–893, June 2005.

[2] BIWI Walking Pedestrians video data set. `http://www.vision.ee.ethz.ch/datasets/index.en.html`.

[3] CAVIAR pedestrians video data set. `http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/` or `http://www-prima.inrialpes.fr/PETS04/caviar_data.html`.

[4] P. Dollar, C. Wojel, B. Schiele, and Perona. P. Pedestrian Detection: An Evaluation of the State of the Art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):743–761, April 2012.

[5] Weina Ge and Robert T. Collins. Vision-Based Analysis of Small Groups in Pedestrian Crowds. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(5), May 2012.

[6] Martin Wirz, Mikkel Baun Kjaegaard, Sebastian Feese, et al. Towards an online detection of pedestrian flocks in urban canyons by smoothed spatio-temporal clustering of GPS trajectories. *ACM LBSN '11*, November 2011.

[7] J. Candamo, M. Shreve, D. B. Goldgof, D. B. Sapper, and R. Kasturi. Understanding transit scenes: a survey on human behavior-recognition algorithms. *ITSS*, 11:206–224, 2010.

[8] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.Q. Xu. Crowd analysis: a survey. *Mach. Vis. Appl.*, 19(5):345–357, 2008.

[9] B. Ni, S. Yan, and A. Kassim. Recognizing human group activities with localized causalities. *CVPR*, (1-2):1063–1089, 2009.

[10] B. Lau, K. O. Arras, and W. Burgard. Multi-model hypothesis group tracking and group size estimation. *IJSR*, 2(1):19–30, 2009.

[11] A. Hoogs, S. Bush, G. Brooksby, et al. Detecting semantic group activities using relational clustering. *WMVC*, (1-2):1–8, July 2008.

[12] McPhail C. *The Myth of the Madding Crowd*. Aldine de Gruyter, 1991.

[13] M. Moussaid, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics. 5(4), 2010.

[14] W. White. City: Rediscovering the Center. *Doubleday*, 1998.

[15] Still G.K. *Crowd Dynamics*. PhD thesis, Univ. of Warwick, 2000.

[16] E.T. Hall. A System for the Notation of Proxemic Behaviour. *Am. Anthropologist*, 65:1003–1026, 1963.

[17] C. McPhail and R. Wohlstein. Using Film to Analyze Pedestrian Behavior. *Sociological Methods and Research*, 10:347–375, 1982.

[18] Papageorgiou C. and Poggio T. A Trainable System for Object Detection. *2000 International Journal for Computer Vision*, 38:15–33, 2000.

[19] Mohan A., Papageorgiou C., and Poggio T. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:349–361, April 2001.

[20] Viola P., Jones M. J., and Snow D. Detecting pedestrians using patterns of motion and appearance. *Ninth International Conference on Computer Vision*, 1:734–741, 2003.

[21] Mikolajczyk K., Schmid C., and Zisserman A. Human detection based on a probabilistic assembly of robust part detectors. *Eighth European Conference on Computer Vision*, 1:69–81, 2004.

[22] Henry Schneiderman and Takeo Kanade. Object detection using the statistics of parts. *2004 International Journal for Computer Vision*, 56:151–177, 2004.

[23] de Poortere V., Cant J., Van den Bosch B., et al. Efficient pedestrian detection: a test case for svm based categorization. *Workshop on Cognitive Vision*, 2002.

[24] Gavrila D.M. and Philomin V. Real-time object detection for smart vehicles. *Internation Conference on Computer Vision and Pattern Recognition*, pages 87–93, April 1999.

[25] Gregoire Malandain and Celine Fouard. On optimal chamfer masks and coefficients. *Research Report 5566, INRIA*.

[26] Gavrila D.M., Giebel J., and Munder S. Vision-based pedestrian detection: the protector+ system. *IEEE Intelligent Vehicles Symposium*, 2004.

[27] Yoav Freund and Robert E.Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, (55), 1997.

[28] Ronfard R., Schmid C., and Triggs B. Learning to parse pictures of people. *Seventh European Conference on Computer Vision*, 4:700–714, 2002.

[29] Felzenszwalb P. and Huttenlocher D. Efficient matching of pictorial structures. *International Conference on Computer Vision on Pattern Recognition*, pages 66–75, September 2000.

[30] Ioffe S. and Forsyth D. A. Probabilistic methods for finding people. *International Journal on Computer Vision*, 43, 2001.

[31] Gavrila D.M. The visual analysis of human movement: A survey. *CVIU*, 73:82–89, 1999.

[32] Piccardi M. Background subtraction techniques: a review. *2004 IEEE International Conference on Systems, Man and Cybernetics*, 4:3099–3104, October 2004.

[33] Stauffer C. and Grimson W. Learning patterns of activity using real time tracking. *IEEE Trans. Pattern Analysis Machine Intelligence*, 8(22):747–767, 2000.

[34] Gao X., Boult T., Coetzee F., , and Ramesh. Error analysis of background adaption. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 503–510, 2000.

[35] Tamersoy B. Background Subtraction - Lecture notes. *University of Texas at Austin*, September 2009.

[36] Brajesh Patel and Neelam Patel. Motion Detection based on multi-frame video under surveillance systems. *2012 IEEE International Conference on Systems, Man and Cybernetics*, 12, March 2012.

[37] Nan Lu, Jihong Wang, Wu Q.H., and Li Yang. An improved Motion Detection method for real time Surveillance. *2012 IEEE International Conference on Systems, Man and Cybernetics*, February 2012.

[38] Benezeth Y., Emile B., Laurent H., and Rosenberger C. Review and evaluation of commonly-implemented background subtraction algorithms. *19th International Conference on Pattern Recognition*, pages 1–4, December 2008.

[39] Wren C.R., Azarbayejani A., Darrell T., and Pentland A.P. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:780–785, July 1997.

[40] Stauffer C. and Grimson W. Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:246–252, August 1999.

[41] Bouwmans T., El Baf F., and Vachon B. Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey. *Recent Patents on Computer Science*, 1:219–237, November 2008.

[42] Cortes C. and Vapnik V. Support-vector networks. *Machine Learning*, 3(3):273, 1995.

[43] Dimitri A. Lisin, Marwan A. Mattar, Matthew B. Blaschko, Mark C. Benfield, and Erik G. Learned-Miller. Combining Local and Global Image Features for Object Class Recognition. *Computer Vision and Pattern Recognition - Workshops*, page 47, 2005.

[44] kalman gain and group tracking explanation. `http://www.mathworks.com/help/vision/examples/people-tracking.html`.

[45] Hai Tao. Image Analysis and Computer Vision: Object Tracking and Kalman Filtering Notes. *University of Califonia at Santa Curz*.

[46] H.W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[47] Bar-Shalom. Y. and Fortmann. T.E. *Tracking and Data Association (Mathematics in Science and Engineering)*, volume 179. January 1988.

[48] M.K. Pitt and N. Shephard. Filtering Via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, pages 590–591, 2008.

[49] Ko BC, Jeong M, and Nam J. Fast Human Detection for Intelligent Monitoring Using Surveillance Visible Sensors. In *Sensors (Basel, Switzerland)*, volume 14, pages 21247–21257, 2014.

[50] N.D. Bird, O. Masoud, N.P. Papanikolopoulos, and A. Isaacs. Detection of loitering individuals in public transportation areas. *Intelligent Transportation Systems, IEEE Transactions on*, 6(2):167–177, June 2005.

[51] Oliver Sidla, Yuriy Lypetskyy, Norbert Brandle, and Stefan Seer. Pedestrian Detection and Tracking for Counting Applications in Crowded Situations. In *Proceedings of the IEEE International Conference on Video and Signal Based Surveillance*, AVSS '06, pages 70–, 2006.

[52] Thomas Michelat, Nicolas Hueber, Pierre Raymond, et al. Automatic Pedestrian Detection and Counting Applied to Urban Planning. In *Proceedings of the First International Joint Conference on Ambient Intelligence*, AmI'10, pages 285–289, 2010.

[53] O. Sidla, Y. Lypetskyy, N. Brandle, and S. Seer. Pedestrian Detection and Tracking for Counting Applications in Crowded Situations. In *Video and Signal Based Surveillance, 2006. AVSS '06. IEEE International Conference on*, pages 70–70, Nov 2006.

[54] Senem Velipasalar, Ying li Tian, and Arun Hampapur. Automatic counting of interacting people by using a single uncalibrated camera. In *IEEE International Conference on Multimedia and Expo, 2006*, pages 1265–1268, 2006.

[55] Huadong Ma, Chengbin Zeng, and Charles X. Ling. A Reliable People Counting System via Multiple Cameras. *ACM Trans. Intell. Syst. Technol.*, 3(2):31:1–31:22, February 2012.

[56] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.

[57] Forczmanski, Pawel, Kukharev, and Georgy. Comparative analysis of simple facial features extractors. *Journal of Real-Time Image Processing*, 1:239–255, 2007.

[58] Ishii, Idaku, Ichida, et al. 500-fps face tracking system. *Journal of Real-Time Image Processing*, 8:379–388, 2013.

[59] Arnulf B. A. Graf and Felix A. Wichmann. Gender classification of human faces. In *In*, 2002.

[60] S. Handri, S. Nomura, and K. Nakamura. Determination of Age and Gender Based on Features of Human Motion Using AdaBoost Algorithms. *International Journal of Social Robotics*, 3(3):233–241, 2011.

[61] Xin Geng, Zhi hua Zhou, Senior Member, Kate Smith-miles, and Senior Member. Automatic age estimation based on facial aging patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:2234–2240, 2007.

[62] An example of industrial video analytics solution. `http://info.singtel.com/large-enterprise/m2m/solutions/video-analytics/solution?gclid=CK7GydWrkMQCFU4ojgodw7IACg`.

[63] H. Elzein, S. Lakshmanan, and P. Watta. A motion and shape-based pedestrian detection algorithm. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 500–504, June 2003.

[64] K. Kaliyaperumal, S. Lakshmanan, and K. Kluge. An algorithm for detecting roads and obstacles in radar images. *Vehicular Technology, IEEE Transactions on*, 50(1):170–182, Jan 2001.

[65] N. Oliver, B. Rosario, and A. Pentland. A Bayesian Computer Vision System for Modeling Human Interactions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:831–843, August 2000.

[66] S. Gong and T. Xiang. Recognition of Group Activities Using a Dynamic Probabilistic Network. *Proc. IEEE Int'l Conf. Computer Vision*, pages 742–749, October 2003.

[67] Haritaoglu and M. Flickner. Detection and Tracking of Shopping Groups in Stores. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 431–438, December 2001.

[68] X. Naturel and J. Odobez. Detecting Queues at Vending Machines: A Statistical Layered Approach. *Proc. Int'l Conf. Pattern Recognition*, pages 1–4, December 2008.

[69] Guler, Puren, Emeksiz, et al. Real-time multi-camera video analytics system on GPU. *Journal of Real-Time Image Processing*, pages 1–16, 2013.

[70] D. Gatica-Perez. Automatic Nonverbal Analysis of Social Interaction in Small Groups: A Review. *Journal on Image and Vision Computing, special issue on human behavior*, 27:1775–1787, 2009.

[71] A. Hoogs, S. Bush, G. Brooksby, et al. Detecting Semantic Group Activities Using Relational Clustering. *Proc. IEEE Workshop Motion and Video Computing*, pages 1–8, January 2008.

[72] A. French, A. Naeem, I. Dryden, and T. Pridmore. Using Social Effects to Guide Tracking in Complex Scenes. *Proc. IEEE Conf. Advanced Video and Signal Based Surveillance*, pages 212–217, 2007.

[73] J. Jacques Jr., A. Braun, J. Soldera, S. Musse, and C. Jung. Understanding People Motion in Video Sequences Using Voronoi Diagrams. *Pattern Analysis and Applications*, 10:321–332, October 2007.

[74] F. Cupillard, F. Bremond, and M. Thonnat. Tracking Groups of People for Video Surveillance. *Proc. European Workshop Advanced Video-Based Surveillance System*, 2001.

[75] D. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Trans. Automatic Control*, 24(6):843–854, December 1979.

[76] B. Lau, K. Arras, and W. Burgard. Multi-Model Hypothesis Group Tracking and Group Size Estimation. *Int'l J. Social Robotics*, 2(1):19–30, 2010.

[77] Jan Sochman and David C. Hogg. Who Knows Who - Inverting the Social Force Model for Finding Groups. *in Proc. IEEE Conf. Computer Vision Workshops*, pages 830–837, 2011.

[78] Arun Kumar Chandran, Loh Ai Poh, and Prahlad Vadakkepat. Identifying Social Groups in Pedestrian Crowd Videos. *Advances in Pattern Recognition, 2015. ICAPR '15. Eighth International Conference on ,accepted*, April 2015.

[79] Weina Ge, R.T. Collins, and B. Ruback. Automatically detecting the small group structure of a crowd. In *Applications of Computer Vision (WACV), 2009 Workshop on*, pages 1–8, Dec 2009.

[80] Yimeng Zhang, Weina Ge, Ming-Ching Chang, and Xiaoming Liu. Group context learning for event recognition. In *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, pages 249–255, Jan 2012.

[81] R. Datta, Weina Ge, Jia Li, and J.Z. Wang. Toward Bridging the Annotation-Retrieval Gap in Image Search. *MultiMedia, IEEE*, 14(3):24–35, July 2007.

[82] Ming-Ching Chang, N. Krahnstoever, and W. Ge. Probabilistic group-level motion analysis and scenario recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 747–754, Nov 2011.

[83] Canny John. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.

[84] Powers and David M W (2007/2011). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness and Correlation. *Journal of Machine Learning Technologies*, 2, 2011.

[85] G. Lindzey R.W. Brown. *Mass Phenomena: Handbook of Social Psychology*, volume 2. Addison Wesley, 1954.

[86] D. Cartwright and A. Zander. *Group Dynamics: Research and Theory third ed.* Harper, 1968.

[87] C. McPhail. *Withs across the Life Course of Temporary Sport Gatherings*. unpublished manuscript, Univ. of Illinois, 2003.

[88] Johnson N.R. Panic at the Who Concert Stampede: An Empirical Assessment. *Social Problems*, 34:326–373, 1987.

[89] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering Similar Multidimensional Trajectories. *Proc. IEEE Conf. Data Eng.*, pages 673–684, February 2002.

[90] B.T. Morris and M.M. Trivedi. Learning, Modeling, and Classification of Vehicle Track Patterns from Live Video. *IEEE Trans. Intelligent Transport Systems*, 9(3):425–427, 2008.

[91] Julio Barros, James French, Worthy Martin, Patrick Kelly, and Mike Cannon. Using the Triangle Inequality to Reduce the Number of Comparisons Required for Similarity-Based Retrieval. *Proc. of SPIE/IS&T Conf. on Storage and Retrieval for Image and Video Databases IV*, pages 392–403, 1996.

[92] Berry C.C. The kappa statistic. *Journal of the American Medical Association*, page 268(18):2513, 1992.

[93] Article on extent of surveillance camera usage in United Kingdom. `http://www.dailymail.co.uk/news/article-1205607/Shock-figures-reveal-Britain-CCTV-camera-14-people--China.html`.

[94] Report on internet bandwidth usage at the National University of Singapore. `http://www.nus.edu.sg/comcen/gethelp/guide/itcare/bandwidth.htm`.

[95] Petrosino, Alfredo, Miralto, et al. A real-time streaming server in the RTLinux environment using VideoLanClient. *Journal of Real-Time Image Processing*, 6:247–256, 2011.

[96] Industrial software for crowd simulations. `http://http://vision-traffic.ptvgroup.com/en-us/products/ptv-viswalk/`.

[97] Fortune and Steven. Handbook of Discrete and Computational Geometry. chapter Voronoi Diagrams and Delaunay Triangulations, pages 377–388. CRC Press, Inc., Boca Raton, FL, USA, 1997.

[98] Jason Tsai, Natalie Fridman, Emma Bowring, et al. ESCAPES- Evacuation Simulation with Children, Authorities, Parents, Emotions, and Social comparison.

[99] Hall Mark, Frank Eibe, Holmes Geoffrey, et al. The WEKA Data Mining Software: An Update, November 2009.

[100] Wayne Iba and Pat Langley. Induction of One-Level Decision Trees. *Proceedings of the Ninth International Conference on Machine Learning*, pages 233–240, July 1992.

[101] Rosenblatt and Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. *Spartan Books, Washington DC,*, 1961.

[102] Leo Breiman. Random Forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[103] Broomhead D. S.and Lowe D. Radial basis functions, multi-variable functional interpolation and adaptive networks (Technical report). *RSRE. 4148*, 1988.

[104] Town centre video data set. `http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/2009bbenfold_headpose/project.html#datasets`.

[105] A method to split data set into training and testing sets for mahine learning. `http://gerardnico.com/wiki/data_mining/resampling`.

[106] S. Ali and M. Shah. Floor Fields for Tracking in High Density Crowd Scenes. *Proc. European Conf. Computer Vision*, 5(1-14), October 2008.

[107] P. Scovanner and M. Tappen. Learning Pedestrian Dynamics from the Real World. *Proc. IEEE Int'l Conf. Computer Vision*, 2009.

[108] S. Pellegrini, A. Ess K. Schindler, and L. van Gool. You'll Never Walk Alone: Modeling Social Behavior for Multi-Target Tracking. *Proc. IEEE Int'l Conf. Computer Vision*, 2009.

[109] G. Antonini, S.V. Martinez, M. Bierlaire, and J.P. Thiran. Behavioral Priors for Detection and Tracking of Pedestrians in Video Sequences. *Int'l J. Computer Vision.*

[110] R. Mehran, A. Oyama, and M. Shah. Abnormal Crowd Behavior Detection Using Social Force Model. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.

[111] Helbing D, Molnar P, Farkas I J, and Bolay K. Self-organizing pedestrian movement. *Environment and Planning B: Planning and Design*, 28(3):361–383, 2001.

[112] Graham D. Finlayson, Steven D. Hordley, Cheng Lu, and Mark S. Drew. Removing Shadows from Images. In *In ECCV 2002: European Conference on Computer Vision*, pages 823–836, 2002.

[113] Daniel Grest, Jan michael Frahm, and Reinhard Koch. A color similarity measure for robust shadow removal in real time. In *In Vision, Modeling and Visualization*, pages 253–260, 2003.

[114] ClÃ©ment Fredembach and Graham D. Finlayson. Hamiltionian path based shadow removal.

# Appendix A

# Performance of Fast HOG

HOG is a computationally expensive technique, where computations are performed on elementary cells of image regions. A typical video consists of a sequential stream of 25 video frames per second. But HOG can process only four video frames per second (for a typically used 640×480 image resolution), which is very slow. This is because only several cores in a computer's processor, sequentially compute on the numerous cells in the video frames. Until the development of the GPU implementation of HOG (Fast HOG), it was practically not possible to use HOG in real-time applications because of its high computational requirement. With Fast HOG, video frame processing rates of up to 60 fps is possible (equivalent to 2x speed for a 25 fps video) for pedestrian detection with an added cost of an average CUDA based graphic card (a GTX 660 NVidia graphic card). Fast HOG performs the cell computations in a parallel manner using the numerous cores in the GPU (Graphic Processing Unit). Hence, it is able to process up to 60 frames in a second with the above mentioned graphic card. The detection performance of the Fast HOG is highlighted in this appendix.

The Fast HOG detection results in two types of background environment (simple and complex) are highlighted in Figure A.1. The results show the robustness of this method in dealing with illumination variations and confusers resembling human shape and dress color. Camera location in Figure A.1 (left section) is a sports ground. This location's background is considered to be a simple background as there are no moving non-pedestrian, and pedestrian resembling entities in the region of interest. Camera location

in Figure A.1 (right section) is a pedestrian area in front of a book store and several ATM machines. This location's background is considered to be complex as there are moving non-pedestrians, and pedestrian resembling entities in the region. The red oval highlights a pedestrian resembling entity detected by the Fast HOG. From Figure A.1, it is evident that Fast HOG can handle partial occlusions and detect pedestrians reasonably well in complex background environments. Tests are performed on a wide range of indoor and outdoor camera locations (Figure 3.1). Figure A.2 shows a stationary pedestrian detected by Fast HOG, while the Background subtraction does not detect the person as the background model merges his location with the background. These results outline the performance of Fast HOG in daylight conditions. Pedestrian detection during night time (at outdoor locations) pose a different set of challenges.

Low illumination and pedestrian shadows[1] posed significant challenges to pedestrian detection in night conditions. Background subtraction is affected by these factors and hence is found to be not suitable in night conditions. Unlike Background subtraction, Fast HOG does not detect shadows of pedestrians (Figure A.4). Existing pedestrian detection techniques employ additional image processing techniques [112, 113, 114] to remove shadow detections. Figure A.3 shows Background subtraction and Fast HOG pedestrian detection results under low illumination. These results are for a night vision video feed with very low illumination. Background subtraction detects several pedestrians together as single moving blob while Fast HOG could detect pedestrians who are in the least illuminated regions also. Figs. A.3 and A.4 show the effectiveness of the Fast HOG in comparison to the Background subtraction applied to the same video. The Fast HOG when appropriately tuned[2] is able to perform considerably better than Background subtraction in night conditions. An approach combining the pedestrian detections of the Background subtraction and the Fast HOG is developed and used in this thesis. This approach is explained in Sec. 3.1.1.

[1]Pedestrian shadows occur in day light conditions also but are less prominent compared to the night time situation. This is because the light source during night time is closer to the region of interest, resulting in prominent pedestrian shadows.

[2]The pedestrian detection confidence factor, grouping factor are some of the free parameters which could be tuned. More information is available in [1].


Figure A.1: Fast HOG detection in a simple (left) and complex (right) backgrounds. A false detection is highlighted by a red oval

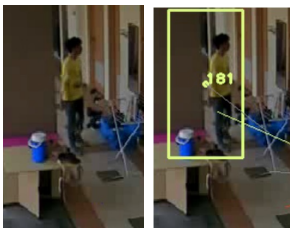


Figure A.2: Stationary person not detected with Background subtraction (left). Fast HOG detects the stationary person (right). Background subtraction does not detect the person, as the background model merges his location with the background.

Figure A.3: Background subtraction (left), Fast HOG (right) pedestrian detections under low illumination.


Figure A.4: Background subtraction (left), Fast HOG (right) pedestrian detections. Moving shadows (red ovals) are detected by Background subtraction while Fast HOG does not detect shadows. Also, stationary pedestrians are not detected by Background subtraction.

# Appendix B

# Spatial distributions learned from Pedestrian Group Record (PGR)

Figures B.1, B.2 and B.3 outline the results of applying spatial distribution learning method to two different queue locations. The technique to learn spatial distribution of pedestrian group members is explained in 6. Figures B.4, B.5 and B.6 outline the people counting results for the two methods discussed for different time periods across five days in a food canteen. It is evident that the automatic queue region detection (by learning the spatial distribution of the queue members) performs on par to the manual queue region marking method.


Figure B.1: Group members' spatial distributions at ten different time periods extracted from the Pedestrian Group Record (PGR), learned spatial distribution and identified queue region (marked by a black polygon) - KA Indian Queue.

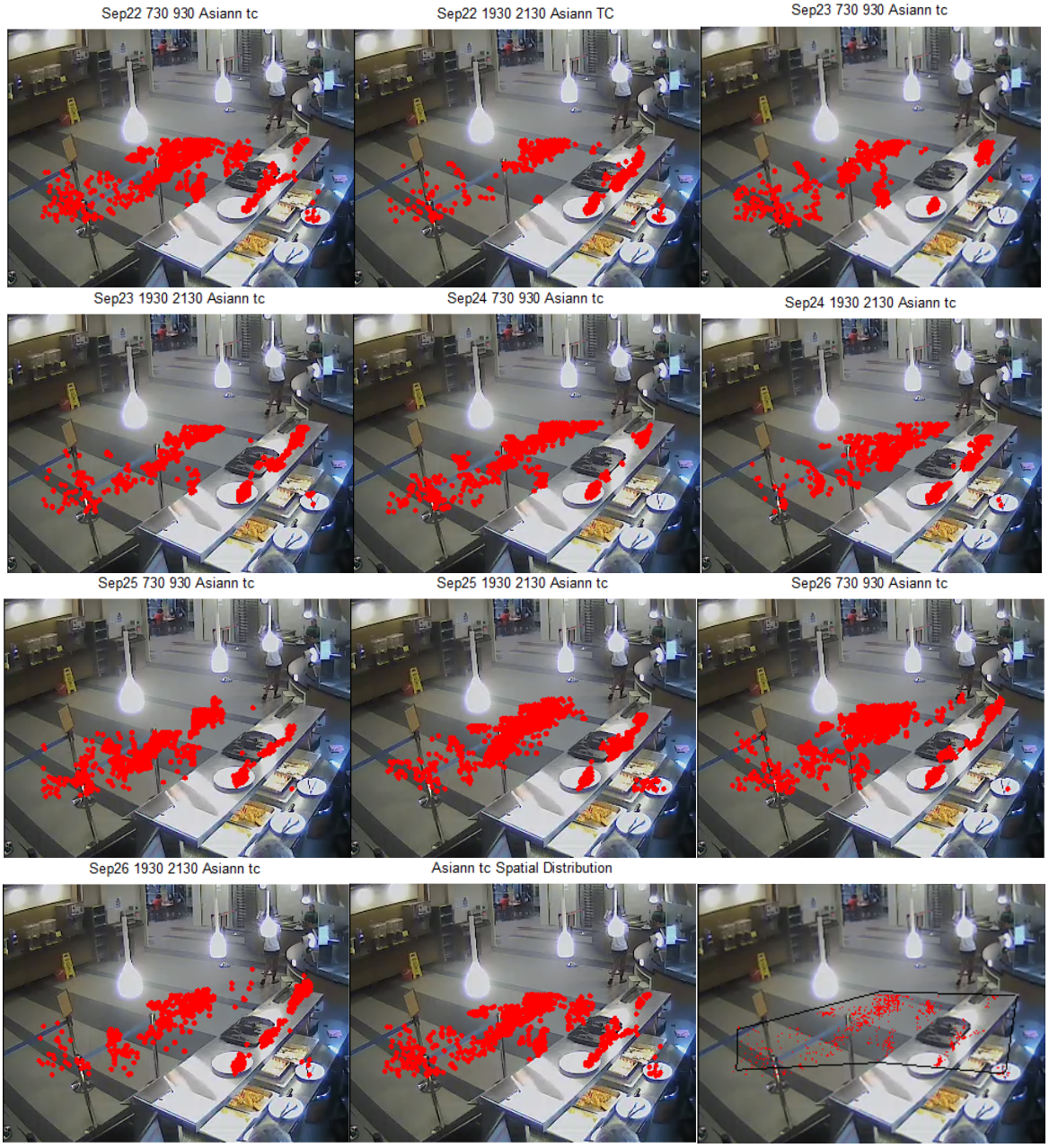



Figure B.2: Group members' spatial distributions at ten different time periods extracted from the Pedestrian Group Record (PGR), learned spatial distribution and identified queue region (marked by a black polygon) - TC Asian Queue.

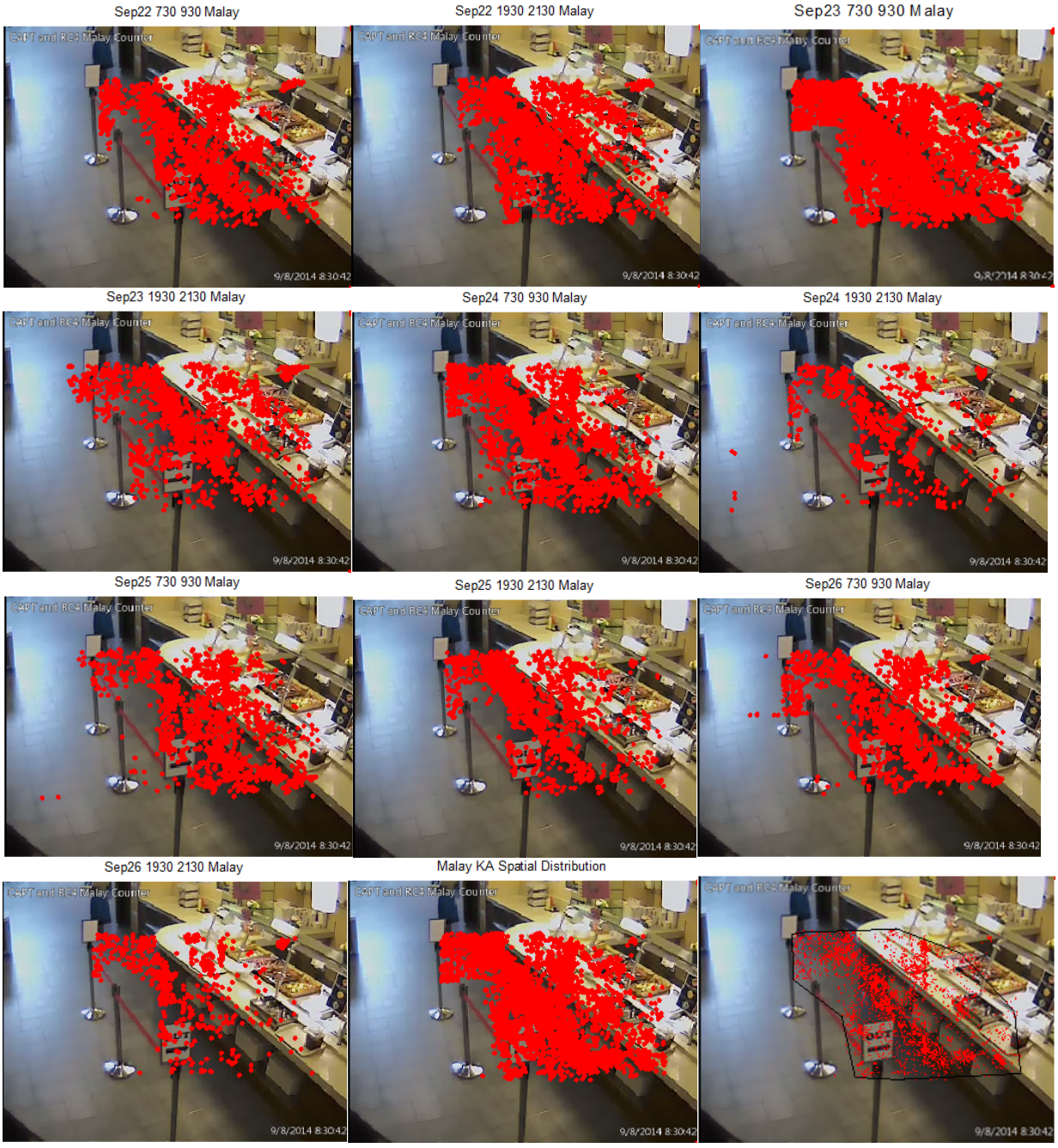



Figure B.3: Group members' spatial distributions at ten different time periods extracted from the Pedestrian Group Record (PGR), learned spatial distribution and identified queue region (marked by a black polygon) - KA Malay Queue.

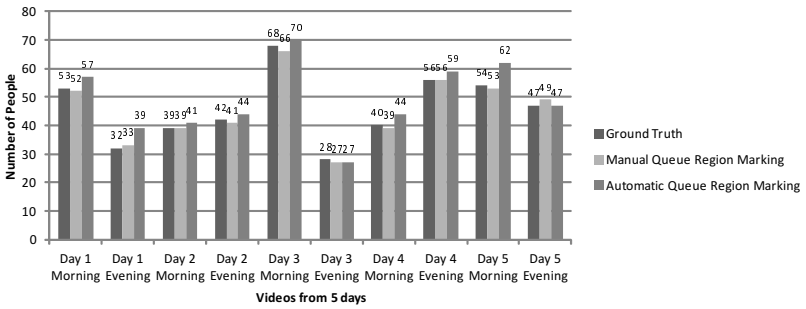



Figure B.4: People counting results for KA Indian queue.

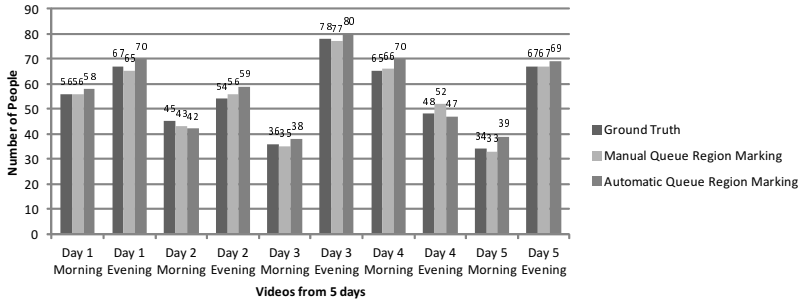



Figure B.5: People counting results for TC Asian queue.

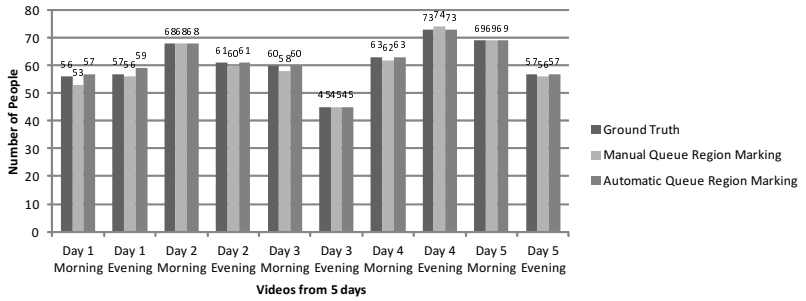



Figure B.6: People counting results for KA Malay queue.

# Appendix C

# Training Methods to Learn Motiom Parameter Variations

Several training methods such as linear SVM, non-linear SVM -Radial Basis Function (RBF), SVM with meta-algorithms such as AdaBoost [27], Decision stump classifier [100], Neural Network [101] and Random Forest [102] were tested. All these standard machine learning techniques are tried to learn the motion parameter variations. The results of these teste are shared in this Appendix. The training time measurements are also shared in this Appendix.

## C.1 Prediction of Potential Customer-Approach to Food Outlets

The food outlet 5 data set was chosen to perform prediction performance testing for different machine learning techniques. This data set was selected as it is considered to be the most complex among all the 5 data sets. The training methods include linear SVM, non-linear SVM -Radial Basis Function, SVM with meta-algorithms such as AdaBoost, Decision stump classifier, Neural Network and Random Forest.

From the above tables, the SVM with meta algorithms was observed to perform better than the other training methods. Hence, these SVM based training methods were utilized

Table C.1: Prediction Performance for different training methods.

| Training Method | Motion Parameters | Linear SVM | SVM - RBF | Linear SVM AdaBoost | SVM - RBF Adaboost | Decision Stump Classifier | Random Forest | Neural Network |
|---|---|---|---|---|---|---|---|---|
| Prediction Performance (%) | Basic | 54.13 | 52.03 | 58.75 | 51.53 | 52.06 | 54.03 | 43.49 |
| | Extended | 75.34 | 60.43 | 80.19 | 57.59 | 55.17 | 67.13 | 49.37 |

Table C.2: Training time measurements for different training methods.

| Training Method | Motion Parameters | Linear SVM | SVM - RBF | Linear SVM AdaBoost | SVM - RBF Adaboost | Decision Stump Classifier | Random Forest | Neural Network |
|---|---|---|---|---|---|---|---|---|
| Training Time (mins) | Basic | 62 | 73 | 64 | 82 | 87 | 71 | 128 |
| | Extended | 69 | 81 | 73 | 93 | 97 | 79 | 142 |

Table C.3: Training time measurements for different training methods.

| Training Method | Linear SVM | SVM - RBF | Linear SVM AdaBoost | Decision Stump Classifier | Random Forest | Neural Network |
|---|---|---|---|---|---|---|
| Training Time (mins) | 55 | 84 | 63 | 103 | 75 | 137 |

to do performance analysis on all the food outlet data sets. This performance analysis is shared in Table 7.1.

## C.2 Prediction of Future Pedestrian Groups

The prediction performance utilizing the different machine learning techniques is performed in all the 6 data sets. These training methods include linear SVM, non-linear SVM -Radial Basis Function, SVM with meta-algorithms such as AdaBoost, Decision stump classifier, Random Forest and Neural Network. The following figure depicts the prediction performance of a model developed from one data set applied to other data sets. A value of 1 (represented by dark red color) indicates 100% accuracy.

From the table C.3 and Figure C.1, the linear SVM is observed to have higher prediction accuracy than the other training methods, taking lesser training time.

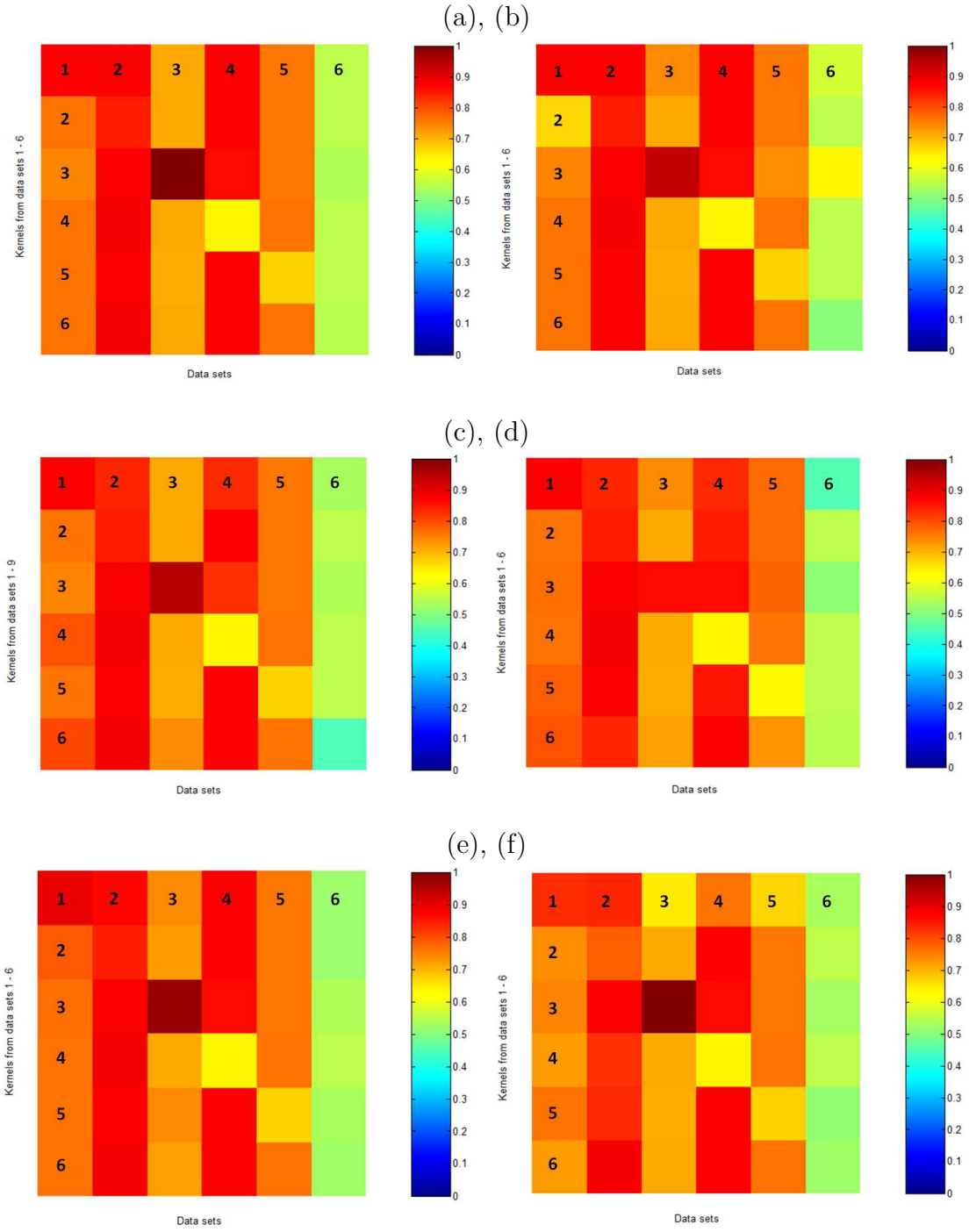



Figure C.1: Prediction performance for data sets 1 to 6 for different training methods. The heat map indicates percentage of prediction accuracy. A value of 1 (represented by dark red color) indicates 100% accuracy. (a) Linear SVM, (b) Non-linear SVM -Radial Basis Function, (c) Linear SVM with AdaBoost, (d) Decision stump classifier, (e) Random Forest and (f) Neural Network.

# Appendix D

# List of Publications

1) Arun Kumar Chandran, Loh Ai Poh, Prahlad Vadakkepat. "Identifying social groups in pedestrian crowd videos," 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR) , vol., no., pp.1,6, 4-7 Jan. 2015

2) Arun Kumar Chandran, Loh Ai Poh, and Prahlad Vadakkepat. “Real-time Identification of Pedestrian Meeting Events from Surveillance Videos using Motion Similarity,” Journal of Real-Time Image Processing, submitted.

3) Arun Kumar Chandran, Loh Ai Poh, and Prahlad Vadakkepat. “Pedestrian Activity Prediction By Learning Pedestrian Motion Patterns,” Journal of Real-Time Image Processing, in preparation.